



Francisco Pereira Cardoso

Licenciatura em Ciência e Engenharia Informática

Gestão e otimização de dados para a criação de análises informativas

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador: André Santos, Manager,
Unipartner IT Services

Co-orientador: Vítor Duarte, Professor Auxiliar,
Universidade Nova de Lisboa

Júri

Presidente: Doutora Teresa Romão, Professora Auxiliar, FCT UNL
Vogais: Doutor Francisco Azevedo, Professor Auxiliar, FCT UNL
Mestre André Santos, Especialista, Unipartner IT Services



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro, 2019

Gestão e otimização de dados para a criação de análises informativas

Copyright © Francisco Pereira Cardoso, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

RESUMO

Com a constante acumulação de dados ao longo do tempo por parte das empresas, é complexo manter uma estrutura organizada e eficiente de armazenamento e análise dos mesmos, sendo por isso notória a falta de aproveitamento de uma análise informativa relevante dos dados para auxiliar nas tomadas de decisão de negócio. O projeto que se pretende desenvolver enquadra-se nesta temática e é dirigido a um cliente que revela esses problemas, procurando uma resolução dos mesmos.

Inicialmente, é necessário fazer uma análise aos dados armazenados pelo cliente e perceber a forma mais eficiente de os tratar e organizar numa solução analítica, de forma a promover um consumo frequente pelos diversos utilizadores. Além disso, é necessário desenhar o processo de extração, tratamento e carregamento dos dados para a solução e definir e implementar uma arquitetura estruturada, eficiente e escalável que satisfaça as necessidades do cliente.

A solução final tem como objetivo promover um maior e melhor aproveitamento dos dados armazenados, assim como a consulta dos mesmos. Outro grande objetivo passa por reduzir o tempo e esforço necessários para a criação de relatórios de negócio, contribuindo assim para a redução de custos da empresa e aumento da produtividade do negócio.

Palavras-chave: *Business Intelligence*, bases de dados, *Data Warehouse*, processo de *ETL*, solução analítica, modelo tabular, factos, dimensões, *dashboards*

ABSTRACT

With the constant accumulation of data over time by companies, it becomes complex to maintain an organized and efficient storage structure and analysis of this data, being therefore notorious the lack of use of an analysis information to assist in business decision-making. The client to whom the solution will be developed presents these same problems and seeks a solution that will solve them.

Initially, it is necessary to analyze the data stored by the customer and understand the most efficient way to process and organize it in an analytical solution, in order to promote a frequent consumption by the various users. In addition, it is necessary to design the data extraction, transformation and loading process and also design and implement a structured, efficient and scalable architecture that meets the client needs.

By consulting it, the final solution aims to promote a bigger and greater use of the stored data obtained. Another major objective is to reduce the time and effort required to create business reports, thus contributing to reducing company costs and increasing business productivity.

Keywords: Business Intelligence, database, Data Warehouse, ETL process, analytical solution, tabular model, facts, dimensions, dashboards

ÍNDICE

1	Introdução	1
1.1	Contexto e motivação	1
1.2	Problema	2
1.3	Objetivo	3
1.4	Estrutura do documento	3
2	Trabalho relacionado	5
2.1	Big Data	5
2.1.1	Vantagens	6
2.2	Business Intelligence	7
2.2.1	ETL	7
2.3	Data Warehouse	9
2.4	Arquiteturas	9
2.4.1	Arquitetura de Inmon	9
2.4.2	Arquitetura de Kimball	10
2.5	Processamento e análise dos dados	11
2.5.1	Discussão	12
2.6	Ferramentas	13
2.6.1	SQL Server Management Studio	13
2.6.2	SQL Server Data Tools	14
2.6.3	Microsoft Power BI	14
2.7	Actividades preparatórias	15
2.7.1	Arquitetura proposta	16
2.7.2	Processo de ETL	16
2.7.3	Resultados finais e contribuições	17
3	Requisitos e proposta de solução	19
3.1	Requisitos de alto nível	20
3.1.1	Melhorias sobre os modelos atuais	20
3.1.2	Implementação de novos modelos	24
4	Implementação	31
4.1	Arquitetura dos novos modelos	31

4.1.1	Vantagens	33
4.2	Modelo Order to Cash	34
4.2.1	Identificação e mapeamento dos dados	34
4.2.2	Extração	36
4.2.3	Transformação	39
4.2.4	Carregamento	39
4.2.5	Modelo tabular	41
4.3	Resultado e avaliação	42
5	Conclusões	45
	Bibliografia	47
I	Código de construção dos objectos analíticos	51

INTRODUÇÃO

Neste capítulo será feita uma primeira abordagem à área onde se insere a dissertação e explicada a motivação da mesma, além do problema conducente do projeto, dos objetivos definidos e soluções esperadas.

1.1 Contexto e motivação

Nos últimos anos tem-se assistido a um número cada vez maior de dados que são criados, geridos e guardados em meios informáticos por parte de inúmeras entidades empresariais. Eric Schmidt, ex CEO da Google, chega a afirmar que "a cada dois dias criamos tanta informação como a que foi criada até 2013, o que equivale a cerca de 5 exabytes de dados"[1].

A contribuir para este crescimento exponencial estão essas mesmas entidades, desde start-ups até grandes empresas, que procuram cada vez mais soluções práticas e eficazes à manutenção, gestão e correta análise dos seus dados, para daí obterem benefícios diretos para as tomadas de decisão nos seus negócios. A digitalização e a informatização de cada vez mais processos foi sendo vista como a solução mais viável para as necessidades crescentes das empresas. As ofertas de Cloud e as tecnologias para o processamento de Big Data tornaram possível acompanhar o crescimento de dados e permitir uma gestão significativamente mais eficiente comparando com as soluções utilizadas até à altura.

No entanto, além da necessidade de armazenamento e gestão dos dados, começou a surgir também a oportunidade de efetuar análises mais complexas aos mesmos. Um estudo realizado no âmbito da perceção sobre a intenção das empresas ao analisar os dados armazenados, mostra que 50 por cento das empresas usa os dados analisados para conhecer os seus clientes, sendo que outros 50 por cento usa para encontrar novos clientes e desenvolver produtos novos. Ainda assim, 60 por cento dos entrevistados respondeu

que não confia totalmente nos dados recolhidos e por eles analisados[2]. Isto deve-se a vários fatores. Um deles centra-se em como os dados são guardados e modelados, sendo que uma má gestão dos dados pode causar impactos significativos no negócio. Além disso, soluções que não têm em conta o volume de dados que vão sendo criados e que não se adaptam ao crescimento de uma forma dinâmica, tornam-se soluções ineficientes. Por outro lado, uma boa estruturação de como os dados são guardados implica uma maior produtividade e consequentemente uma maior qualidade nas análises feitas com base nos mesmos [3].

O projeto foi realizado na Unipartner IT Services, uma consultora tecnológica e de integração de sistemas de informação, constituída por uma equipa multidisciplinar e diversificada de profissionais distribuídos pelas diversas áreas. A Unipartner é um dos principais parceiros empresariais da Microsoft e apresenta aos seus clientes soluções construídas maioritariamente usando as suas tecnologias. O projeto em questão enquadra-se na área de *Data & AI/Business Intelligence* e é dirigido a um cliente no setor da indústria da transformação de cortiça que pretende uma solução central analítica e de consumo de informação de um grande volume de dados que tem vindo continuamente a armazenar, de forma a dar apoio à tomada de decisão na área de informação operacional e comercial.

1.2 Problema

Com a constante acumulação de dados ao longo do tempo, torna-se crucial ter atenção a diversos pontos quanto à gestão e análise dos mesmos, para que estes tenham algum valor para a empresa. É neste momento que nos deparamos com vários problemas e desafios referentes ao desenho e desenvolvimento de uma solução de Business Intelligence.

Tipicamente no desenho de uma solução de gestão e análise de dados, é necessário definir o processo desde a fonte dos dados até à fase final de amostragem[4]. Uma das componentes principais neste processo é o Data Warehouse(DW), visto ser a componente responsável pelo armazenamento de toda a informação. Neste ponto, existem algumas escolhas a ser feitas quanto à forma de armazenamento e distribuição dos dados nas várias tabelas que compõem a base de dados e de como os dados se interligam entre si, de forma a minimizar o tempo de resposta a pesquisas feitas, eliminando quaisquer hipóteses de perda de informação e potencializando as análises realizadas. Outra componente da solução é toda a componente de extração, tratamento e carregamento dos dados no Data Warehouse. Existem algumas preocupações na implementação dessa fase, como por exemplo ao nível da extração em que é necessário realizar o correto mapeamento e nomeação dos campos da fonte para o DW, de forma a não adulterar os dados. Ao nível do tratamento dos dados, um dos pontos importantes é compreender de que forma se realiza esse mesmo tratamento, pois diferentes decisões vão influenciar os resultados obtidos, como o aumento no tempo de execução do processamento diário dos dados. Ao nível da arquitetura em geral, existem várias decisões a ser tomadas de forma a dar suporte a toda

a solução sem que isso resulte em custos desnecessários ao cliente mas que seja facilmente gerível e escalável.

O principal problema que o cliente apresenta é a falta de uma solução centralizada eficiente de armazenamento de dados que permita uma análise fiel aos mesmos de forma a criar valor para algumas áreas da empresa, como por exemplo para as áreas comercial e operacional. Como resultado disso, o cliente armazena os dados mas não dispõe de um acesso fácil aos mesmos, não tem uma solução dinâmica que permita obter uma visão mais geral ou mais específica dependendo da intenção, tem dificuldade em manter um acesso à consulta dos dados organizada e atualizada e tem dificuldade em coordenar a análise dos dados de diferentes fontes. Assim, o cliente não está a conseguir que os seus comerciais estejam focados nas principais oportunidades de negócio que possam surgir da relação com os seus clientes, além de existirem oportunidades de análise que com a solução atual não são suportadas.

1.3 Objetivo

O objetivo desta dissertação é desenhar e implementar uma solução central analítica de dados, ou seja, desenvolver uma solução de Business Intelligence para o cliente já referido, de forma a sistematizar a informação que o cliente já possui e criar valor a partir dessa mesma informação. Mais detalhadamente, pretende-se melhorar componentes na solução analítica já existente, composta por três modelos: finanças, stocks e compras. Adicionalmente, pretende-se estender essa mesma solução com a criação de novos modelos, nomeadamente um modelo de *Order to Cash*, que engloba todos os dados relativos às vendas práticas, um de *Purchase to Pay*, que engloba todos os dados relativos às compras e outro de Transportes.

Espera-se que a implementação desta solução traga vários benefícios para o cliente, tais como a melhoria do acesso à informação, a diminuição do tempo na produção de relatórios e mapas, a simplificação do acesso à informação analítica do negócio e de uma forma geral o aprimoramento da organização dos dados da empresa. Adicionalmente, uma evolução para ser possível realizar análises preditivas aos dados é uma ambição para o cliente na medida em que aumenta a possibilidade de análises mais avançadas.

1.4 Estrutura do documento

O documento é composto por mais quatro capítulos além deste. Um segundo capítulo em que são descritos vários conceitos de Business Intelligence e são apresentadas as ferramentas que serão utilizadas no desenvolvimento do projeto, além de serem descritas algumas atividades preparatórias realizadas de adaptação às ferramentas; um terceiro capítulo em que são enunciados os requisitos específicos do cliente e apresentada a proposta de resolução desses requisitos, além de se justificar as escolhas efetuadas; um quarto capítulo

em que são descritos todos os passos de implementação, incluindo um subcapítulo de avaliação dos resultados alcançados e um quinto e último capítulo de conclusões.

TRABALHO RELACIONADO

Apesar de terem objetivos diferentes, a área de Business Intelligence (BI) e de Big Data podem ser vistas como complementares. Enquanto que Big Data reúne as tecnologias necessárias ao processamento de grandes quantidades de dados, Business Intelligence foca-se mais no tratamento e disponibilização dos mesmos para a criação de análises informativas. De forma a perceber melhor ambos os conceitos, este capítulo apresenta algumas características de Big Data, identifica os principais objetivos de Business Intelligence, descreve a importância de uma Data Warehouse, identifica arquiteturas existentes na construção de uma solução de BI e apresenta tecnologias utilizadas que auxiliam no desenvolvimento das soluções, além de demonstrar quais as suas principais vantagens e limitações. O último subcapítulo apresenta algumas atividades preparatórias realizadas que aplicam alguns dos conceitos e ferramentas descritas.

2.1 Big Data

Big Data pode ser definida como "a área dedicada à análise, processamento e manutenção de um grande conjunto de dados que frequentemente têm origem em diversas fontes"[5], com o intuito de obter algum tipo de informação. Devido ao grande volume de dados envolvido neste tipo de processamentos, habitualmente a execução dos algoritmos e a manutenção dos dados é efetuada na Cloud em serviços como por exemplo a Amazon Web Services ou o Microsoft Azure, disponibilizados pela Amazon e a Microsoft, respetivamente. Com soluções em Cloud, torna-se fácil fazer a gestão dos recursos que estão a ser utilizados, os mesmos são facilmente escaláveis e os preços são adaptados às necessidades dos utilizadores, que pagam apenas o que estão a usar. A informação armazenada, que vai sendo constantemente atualizada, pode posteriormente ser utilizada para inúmeros fins,

como por exemplo para o melhoramento de serviços, oferta de novos produtos ou simplesmente para obter uma visão mais integral da área do negócio em análise. Posto isto, é crucial compreender que com a constante acumulação dos dados nas infraestruturas e com a crescente diversidade de frameworks de processamento distribuídas à disposição, o foco de Big Data não se centra apenas na possibilidade de guardar e gerir grandes volumes de dados mas em como utilizá-los de forma relevante e que acrescentem valor.

Do ponto de vista das características que compõem Big Data, o conceito pode ser dividido em 3 grandes "v's"[6]: **volume**, que caracteriza a dimensão de dados que hoje em dia é possível armazenar e manter estável geralmente em soluções na Cloud; **velocidade**, que quantifica a velocidade a que esses dados são produzidos diariamente e carregados para as infraestruturas dos serviços, que têm de ser flexíveis ao ponto de não criar obstrução ao fluxo normal de entrada e por último **variedade**, que se refere à heterogeneidade dos dados que são gerados e que têm de ser guardados. Independentemente de estarem estruturados ou não, de serem imagens, vídeos ou voz, todos os dados têm de ser passíveis de ser guardados de forma a que seja possível uma análise futura. Além do volume, velocidade e variedade, vários autores caracterizam Big Data com ainda mais V's, como é o caso da SAS, empresa tecnológica com grande atuação na área de análise de dados, que adiciona mais 2 ao conjunto[6], George Firican que chega a 10[7] ou mesmo os 42 V's enumerados por Tom Shafer[8], numa intenção de mostrar a capacidade e o rápido crescimento que Big Data teve nos últimos anos e das inúmeras áreas que pode abranger e influenciar positivamente.

2.1.1 Vantagens

De um ponto de vista empresarial, as vantagens do uso de Big Data afetam positivamente vários aspetos do que é o negócio e os objetivos de uma empresa. Como já acima referido, estas vantagens não são alcançadas pelo facto de ser possível guardar um grande volume de dados mas em como utilizar estes dados para retirar valor dos mesmos. Ao construir uma solução que analise corretamente estes dados, os resultados podem conduzir para situações que permitam reduzir custos ou mesmo criar produtos novos.

Um estudo realizado pela NewVantage Partners[9] lista algumas das principais vantagens do uso de Big Data que as empresas envolvidas no inquérito identificaram, entre as quais:

- Auxílio à tomada de decisão - a análise dos dados processados e guardados da empresa ao longo do tempo podem dar respostas que ajudam quanto aos próximos passos a ser tomados;
- Aumento de produtividade - ferramentas que processam grandes quantidades de dados em sistemas distribuídos de forma flexível e eficiente aumentam a resposta face às necessidades das empresas;

- Agilidade - com a oportunidade de controlar constantemente os dados que vão sendo gerados, uma das grandes vantagens que as empresas identificaram é a facilidade com que se identificam oportunidades de mudanças sem que isso produza resultados inesperados.

Além destas, existem inúmeras outras vantagens que justificam a crescente adesão por parte das empresas a técnicas de Big Data e soluções que suportam os seus negócios nestas técnicas.

O escopo desta tese irá envolver todo o processo desde o mapeamento de dados na sua fonte até à disponibilização dos mesmos numa solução analítica, domínio de uma área específica de Big Data - Business Intelligence.

2.2 Business Intelligence

O termo de BI - *Business Intelligence* - surgiu com o intuito de definir todo o processo que engloba o tratamento, armazenamento e análise dos dados de forma a criar análises informativas relevantes para serem apresentados a responsáveis de qualquer área que necessitem de ter uma visão mais detalhada desses mesmos dados, ou, se pretendido, uma visão mais geral. Para o efeito, existe uma estruturação prévia da forma em como os dados irão ser organizados num contexto que faça sentido ao negócio. Depois, durante cada fase do processo, são usadas diversas tecnologias existentes que facilitam o controlo e a interligação entre essas mesmas fases. De uma forma global, uma solução de BI tenta responder a várias perguntas como "Quando?", "Quanto?" e "O quê?" e é por via dessas ferramentas que auxiliam à modelação, gestão, construção de visualizações, gráficos, análises detalhadas de métricas, desvios e tabelas que as respostas a essas perguntas se tornam possíveis. Na fig 2.1 é possível ter uma visão geral do processo desde a recolha de dados até à fase final de análise.

Analisando de uma forma mais granular, o fluxo normal dos dados numa solução de BI passa por 3 fases principais: a extração dos dados das fontes e o seu tratamento, o armazenamento numa Data Warehouse e a saída sob a forma de informação relevante.

2.2.1 ETL

Numa fase inicial, os dados têm de ser recolhidos das diversas fontes e têm de ser tratados de forma a ficarem organizados de forma homogênea, para posteriormente serem inseridos de forma consistente na Data Warehouse. Esta é a fase de ETL (2.2), que significa "extract", "transform" e "load", ou seja, extração transformação/tratamento e carregamento.

A **extração** é responsável por selecionar os dados das diversas fontes e disponibilizá-los de forma agrupada para a fase seguinte. No projeto a ser desenvolvido, a maioria dos dados estão armazenados num sistema integrado de gestão empresarial e a mesma contém tabelas com nomeações internas. Nesta fase, é fundamental mapear corretamente

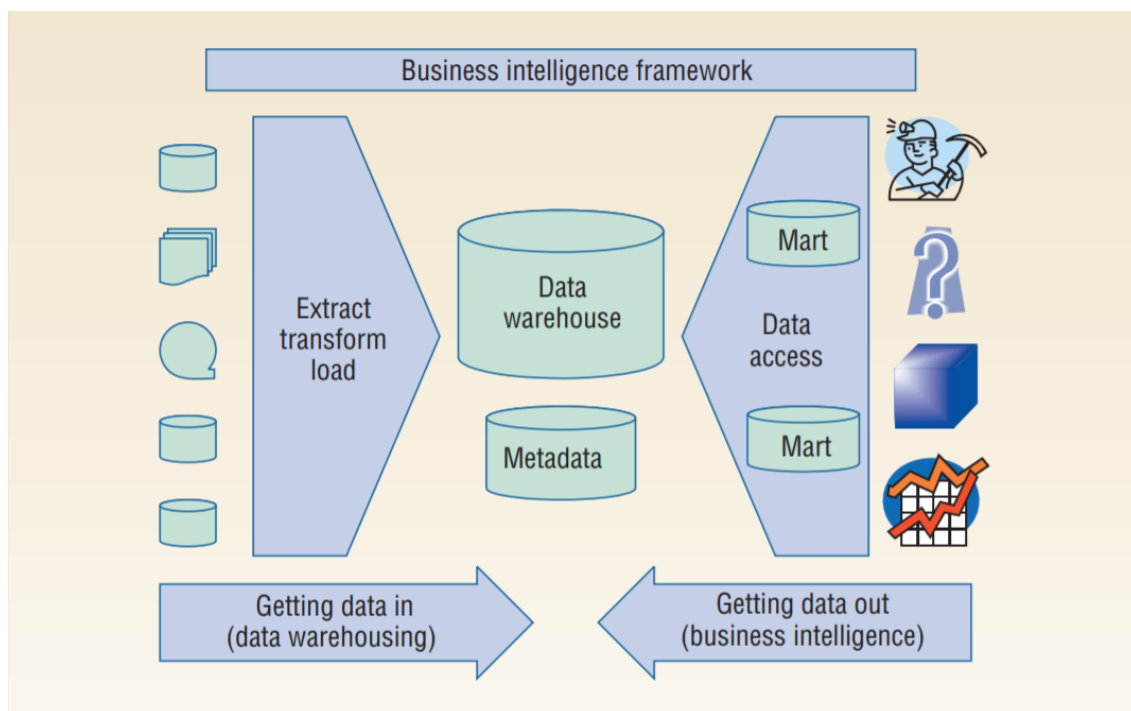


Figura 2.1: Diagrama de uma solução de Business Intelligence [10]

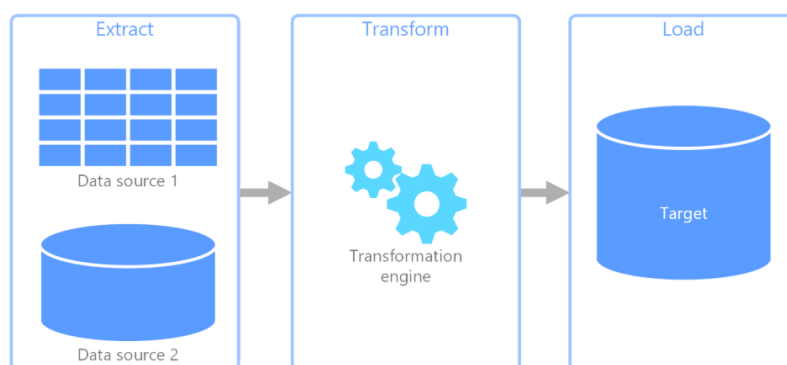


Figura 2.2: Processo de ETL [11]

os campos antes da extração e perceber que tipo de dados vão ser tratados no contexto do negócio.

Na fase de **transformação** e tratamento, é feita uma revisão dos dados e da forma em como estes estão guardados. A ideia principal nesta fase é a de começar a idealizar o esquema relacional responsável pelo armazenamento dos dados, definindo os seus tipos, as relações entre as várias tabelas e realizando outros tipos de verificações, como por exemplo a eliminação de tuplos repetidos não necessários, o tratamento de campos mal preenchidos e a resolução de problemas relacionados com violações de restrições de integridade. Durante o desenho do esquema que servirá de base à organização dos dados no Data Warehouse, é comum realizar uma distinção entre dois tipos de dados que são guardados em dois tipos de tabelas diferentes (no caso do mesmo se basear num modelo

multidimensional): as tabelas de dimensão e as tabelas de facto. No subcapítulo 2.4.2 será apresentada a arquitetura que se apoia nesta divisão de tipos e explicada a sua diferença e no fim deste capítulo irá ser descrito um exemplo de aplicação deste tipo de tabelas.

Por último, existe a fase de **carregamento** dos dados para a Data Warehouse. Esta fase garante que, após a fase de tratamento e esquematização dos dados, estes ficam guardados de forma consistente em uma ou mais bases de dados relacionais e que ficam acessíveis para serem analisados.

2.3 Data Warehouse

Um Data Warehouse pode ser visto como o ponto central do fluxo de dados desde os seus ficheiros de origem até às suas análises e é aí que os dados são mantidos e disponibilizados para consulta sob a forma de dashboards e de relatórios. Existem vários benefícios de se ter um armazém de dados numa solução de BI[12] [13]. Um deles é o de se saber a estrutura que foi definida para os dados e saber como estes estão guardados nas bases de dados, facilitando a forma como se faz o acesso e tornando os dados guardados mais fidedignos. Outro benefício é referente à rapidez de acesso, que é possível graças à centralização dos dados num único armazém instanciado na Cloud e disponível em qualquer momento. Além disso, de um ponto de vista analítico, é interessante ter acesso a uma evolução histórica dos dados, o que é possível com a Data Warehouse devido à capacidade de manter grandes volumes de dados e de os manter persistentes.

2.4 Arquiteturas

Apesar de todas as implementações de soluções de Business Intelligence terem o mesmo fim - o de produzir valor para a empresa em forma de análises aos dados - a construção da solução pode ter várias variantes quanto à sua arquitetura. Existem dois tipos de arquiteturas adotadas mais frequentemente quando se tenciona modelar uma solução - a arquitetura de Inmon e a arquitetura de Kimball. Ambas têm as fases já descritas que compõem o processo de ETL e o carregamento é efetuado para um armazém de dados central, mas existem diferenças na abrangência de alguns processos e em como os dados são organizados e consumidos[14].

2.4.1 Arquitetura de Inmon

Bill Inmon é por muitos considerado o pai do "Data Warehousing" e escreveu várias obras e artigos[15] [16] no qual expõe a sua abordagem quanto à organização de um Data Warehouse, quais as componentes envolvidas e como se ligam entre si. Para Inmon, o armazém de dados é a componente central ao fluxo dos dados e estes são posteriormente agrupados em Data Marts para serem apresentados aos utilizadores, usualmente orientados ao departamento que necessita de aceder àquele conjunto de dados. Data Marts

são fundamentalmente conjuntos agrupados de dados que se relacionam entre si no contexto da empresa, ou seja, são subconjuntos separados de dados do Data Warehouse e são apresentados apenas a quem os precisa de consumir[17]. Uma visão mais geral da metodologia de Inmon pode ser vista na imagem 2.3.

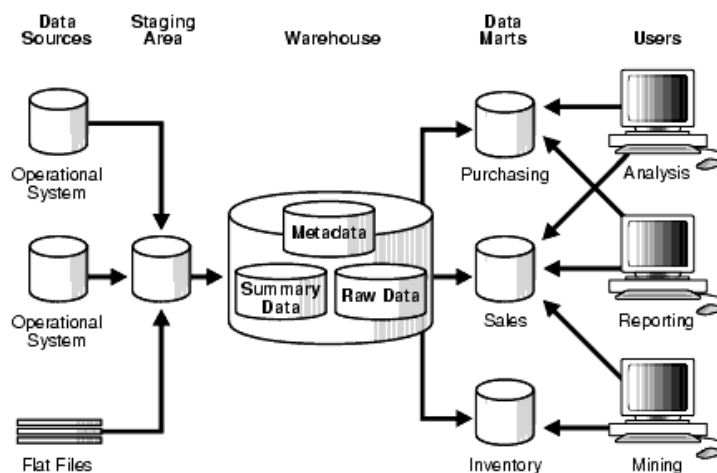


Figura 2.3: Arquitetura de Inmon [18]

Numa fase inicial de modelação, é necessário definir as entidades que constituem o modelo do negócio. Estas entidades irão agrupar toda a informação relacionada entre si e irá permitir criar relações com outras entidades. A ideia principal é que todos os dados que sejam carregados para a Data Warehouse sejam integrados numa entidade, adaptando-se ao modelo desenhado e permitindo manter sempre bem definida a estrutura do negócio. Quanto ao acesso a estes dados, os mesmos são feitos aos Data Marts criados que adicionam um nível na organização do esquema e aumentam a velocidade de resposta às *queries* que por sua vez já são orientadas a estes subconjuntos de dados. Além disso, os modelos de entidades criados nos sistemas de gestão das bases de dados relacionais numa arquitetura de Inmon são normalizados, com o intuito de eliminar possíveis redundâncias nos dados.

2.4.2 Arquitetura de Kimball

Kimball foi também responsável pelo desenvolvimento do conceito de Data Warehousing e propôs um novo modelo procedimental, conhecido como modelo dimensional. Num modelo dimensional os dados são divididos em 2 tipos de tabelas: as tabelas que guardam as dimensões e as tabelas que guardam factos [19]. Uma dimensão pode ser vista como um objeto no contexto do projeto e um facto como uma ocorrência ou uma medida desse objeto num dado instante. Por exemplo, num contexto de vendas de uma loja, uma das dimensões existentes pode ser referente aos artigos, que têm como atributos um nome, uma marca, um modelo e uma secção. Do ponto de vista cronológico, um produto mantém-se de dia para dia. Os factos podem ser todas as ocorrências de venda daquele

produto, que guarda a data, hora, localização e o responsável da venda. Neste caso, os tuplos desta tabela serão únicos, pois só haverá uma venda daquele produto, àquela hora e naquela localização. No esquema de base de dados relacional, uma dimensão pode ter vários factos, o que corresponde a uma relação de um para muitos. Um facto por sua vez terá uma chave estrangeira que referencia uma dimensão (chave primária) e por isso corresponde apenas a uma dimensão, ou seja, uma relação de muitos para um.

Quanto à esquematização destas tabelas de factos e dimensões, as mesmas podem também ter duas metodologias associadas: uma esquematização em estrela ou em floco de neve.

- **Esquema em estrela**

Um esquema em estrela assemelha-se estruturalmente a uma estrela na medida em que existe uma tabela factual no epicentro que liga os seus atributos pela sua chave estrangeira a várias tabelas de dimensão posicionadas à sua volta. Sendo assim, não existirão tabelas de dimensão ligadas umas às outras e os atributos de cada uma estarão mais agregados. Isso implica que neste tipo de soluções é mais intuitivo analisar a implementação mas por outro lado não existe uma normalização das tabelas de facto.

- **Esquema em floco de neve**

Um esquema em floco de neve também apresenta uma factual no centro e várias dimensões dispostas em redor. No entanto, a grande diferença reside no facto das dimensões serem normalizadas, o que origina mais tabelas de dimensões no esquema dispostas em redor das já presentes. Isto reduz a redundância de informação presente nas *queries* à base de dados, mas implica um maior particionamento das tabelas[20].

2.5 Processamento e análise dos dados

Enquanto que as diferentes metodologias e esquemas de dados representam a forma como os mesmos são armazenados no *DW*, surgem também diferentes alternativas quanto à forma de modelar e aceder a estes mesmos dados. Para o efeito, surgem as tecnologias OLAP (*online analytical processing*)[21]. As tecnologias OLAP são responsáveis pela modelação/organização dos dados armazenados de forma a possibilitarem consultas mais rápidas e complexas, sem que isso afete os dados já armazenados no *DW* e fornecendo várias agregações e valores pré-calculados que aceleram o acesso a informações consultadas regularmente.

Como resultado disso, surge a oportunidade de criação de modelos semânticos que contextualizam os dados do cliente, criando relações que criem uma visão perceptível do negócio para os utilizadores. Existem dois tipos de modelos semânticos que podem

ser criados: modelos multidimensionais, como cubos OLAP, ou modelos bidimensionais, como modelos tabulares.

Os cubos OLAP são estruturas multidimensionais que permitem aumentar o poder analítico feito aos dados, na medida em que agregam e pré-calculam valores e tornam a análise mais rápida[22]. Os cubos têm de ir sendo constantemente atualizados para refletir a versão mais atual dos dados, visto que cada momento corresponde a um *snapshot* dos dados da última atualização feita.

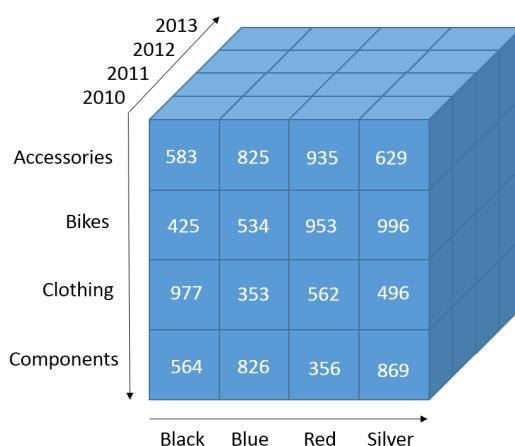


Figura 2.4: Exemplo de um cubo OLAP [23]

Na figura 2.4 podemos ver um exemplo de um cubo OLAP, com três dimensões distribuídas pelos três eixos e com uma medida calculada para cada posição. O tempo de resposta a uma *query* ao valor que seja a interseção destas três dimensões será menor comparativamente a uma situação em que o cálculo ainda tivesse de ser processado.

Os modelos tabulares são estruturas bidimensionais que se baseiam nas bases de dados relacionais já pré-definidas no *DW*. Permite a fácil gestão dos dados que são carregados no modelo e das métricas que se podem criar para as várias tabelas existentes, potencializando também o uso de diferentes formas de visualização dos dados, como por exemplo *Pivot Tables* no Excel.

2.5.1 Discussão

O uso de modelos dimensionais tem sido cada vez a opção mais viável por parte das empresas para a gestão dos seus dados. A razão deve-se a duas características deste tipo de modelos[24]: a de esquematizar os dados de forma intuitiva para a interpretação de quem necessite em qualquer área no meio empresarial e a de retornar rapidamente os dados pretendidos quando são feitas as *queries* aos modelos. Quanto ao esquemas que organizam os dados e os relacionam entre si dentro dos modelos dimensionais, é necessário ter outras considerações, tais como a normalização e a complexidade das *queries* [25]. Relativamente à normalização, um esquema em floco de neve, como já mencionado, terá as suas tabelas

normalizadas e por isso irá eliminar tuplos redundantes. Em larga escala, isso irá reduzir no espaço necessário ao armazenamento das tabelas de dimensão mas será necessário um maior número de tabelas no esquema. Quanto às *queries* feitas aos dois tipos de estruturação, existe uma complexidade acrescida aos esquemas em floco de neve devido à maior partição das tabelas. Isto traduz-se concretamente num maior número de *joins* necessários quando se pretende devolver informação mais detalhada.

De um ponto de vista geral, todo o processo que compõe uma solução de BI, desde a extração dos dados da fonte e o seu respetivo tratamento até à construção de um modelo dimensional para o acesso à informação, tem sido visto como a melhor solução que as empresas encontram para a resolução dos seus problemas relativos ao acesso aos seus dados recolhidos para auxiliar no processo de tomada de decisão. As escolhas de implementação e as ferramentas utilizadas estão dependentes das necessidades do cliente, mas a lógica por detrás da solução final passará, por enquanto, por uma sequência de processos até se obter um modelo central de análise, sendo que o mesmo tem de ser consistente, atualizável e apto a servir de base a todo o tipo de componentes de visualização.

2.6 Ferramentas

Na construção da solução de BI irão ser usadas várias ferramentas que ajudam à esquematização dos dados, integração e à sua análise. A Unipartner, como parceiro estratégico da Microsoft, opta por usar as suas ferramentas no apoio aos seus projetos. Neste projeto em específico, irá ser usado o SQL Server Management Studio para o controlo e acesso das bases de dados, serviços de Cloud Computing Microsoft Azure para carregar e manter as bases de dados, serviços de integração e análise geridos a partir do Visual Studio e o Microsoft Power BI para criar as análises visuais informativas a partir dos dados tratados e organizados.

2.6.1 SQL Server Management Studio

O SQL Server Management Studio (SSMS) é um "ambiente integrado para a gestão de qualquer infraestrutura, desde servidores SQL até bases de dados SQL Azure"[26]. Tem as ferramentas necessárias que auxiliam à configuração e gestão das ligações aos servidores das bases de dados SQL, além de oferecer uma forma intuitiva de gerir as várias componentes que integram o sistema de base de dados.

O SSMS apresenta três zonas principais na sua interface gráfica:

- O **Object Explorer**, que permite visualizar as conexões e de uma forma hierárquica os objetos que vão sendo criados nas bases de dados, como por exemplo tabelas, vistas e procedimentos. Além disso é ainda dada a opção de pesquisar estes objetos criados pelo nome.

- O **editor de queries**, que corresponde à zona onde são desenvolvidos e executados os comandos para a criação dos objetos e permite executar *queries* de acesso às tabelas e vistas.
- A zona de **output** dos comandos, que retorna o resultado da execução dos comandos e apresenta uma listagem dos erros no caso de falha na execução.

2.6.2 SQL Server Data Tools

As ferramentas usadas neste projeto (geridas a partir do Visual Studio) são ferramentas de integração - SQL Server Integration Services (SSIS)[27] e de análise - SQL Server Analysis Services (SSAS)[28].

As primeiras são essenciais para o processamento e integração de dados desde uma fonte até um destino. As fontes de dados provêm de diferentes origens (podendo ser internas ou externas). Essas fontes de dados podem ter diferentes formatos, sendo extraídas, tratadas e consolidadas num ponto central denominado de Data Warehouse (DW). A criação de pacotes SSIS produz o fluxo de ligação necessário entre a fonte e o destino, sendo possível manipular os dados para que sejam armazenados de acordo com o formato pretendido.

Por sua vez, os serviços analíticos SSAS, como o próprio nome indica, são fundamentais para interpretar os dados guardados e poder criar as análises informativas, como por exemplo métricas e indicadores de performance, com recurso à implementação de modelos semânticos. Existem várias linguagens que permitem fazer a extração de informação dos dados, como por exemplo a linguagem MDX, para fazer *queries* aos cubos OLAP ou DAX, que permite pesquisar ao nível das tabelas e criar métricas calculadas.

Ambos os serviços têm funcionalidades adicionais que podem vir a ser utilizadas no decorrer do projeto.

2.6.3 Microsoft Power BI

O Power BI é um serviço analítico de dados desenvolvido pela Microsoft habitualmente mais direcionado ao meio empresarial[29]. A ideia geral é a de construir relatórios em tempo real com várias visualizações dinâmicas de dados de forma a dar uma visão analítica atual sobre o negócio. O Power BI é dividido em 3 componentes principais: o **Power BI Desktop**, o **Serviço Power BI** e as **aplicações móveis**.

O Power BI Desktop é um ambiente de trabalho dedicado à criação de relatórios. Numa fase inicial, é necessário ter acesso aos dados que vão fazer parte desses relatórios importando-os ou realizando *queries* às fontes. As fontes vão desde ficheiros até bases de dados e serviços online. Após ser feito o acesso aos dados, os mesmos ficam acessíveis na aba 'Campos' mais à direita (fig 2.5). Na aba à esquerda - 'Visualizações' - é possível criar relações visuais entre os dados, como gráficos de barras, gráficos circulares, métricas, KPI's com metas a atingir, cartões informativos, entre muitos outros. A combinação de

várias visualizações que "contam uma história" dos dados armazenados da empresa, quer mais global, quer mais específico de uma área, são chamados de "dashboards" e são um dos produtos finais de uma solução de Business Intelligence.

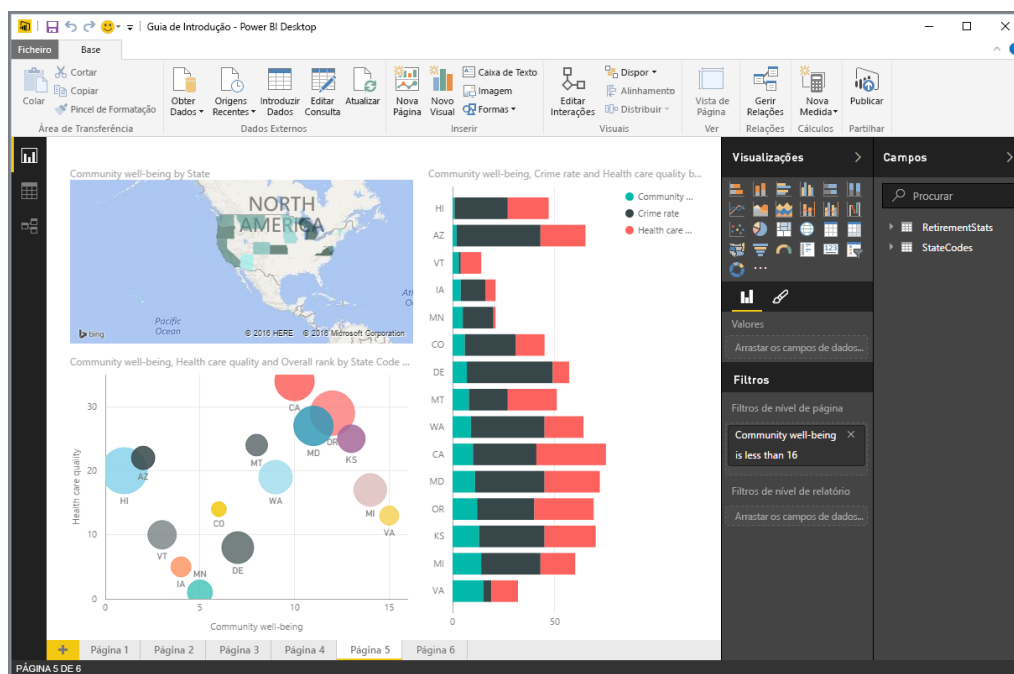


Figura 2.5: Power BI Desktop

O Serviço Power BI e as aplicações móveis são as componentes responsáveis pela partilha e visualização dos *dashboards* criados, existindo sempre a opção de atualizar para ver os dados mais recentes. No caso das aplicações móveis, que permite aos utilizadores verem os dashboards nos seus dispositivos móveis, é necessária a construção manual dos dashboards adaptados às menores dimensões dos dispositivos.

2.7 Actividades preparatórias

Antes de iniciar o desenho e implementação do projeto, foram realizadas algumas atividades de menor dimensão noutro projeto mas com alguma importância pois permitiram uma interiorização inicial de conceitos de Business Intelligente e ter um primeiro contacto prático com o desenho de uma arquitetura, processo de ETL e criação de dashboards.

O conjunto de tarefas realizadas insere-se num projeto piloto para um cliente do setor financeiro que pretendia dinamizar um conjunto de dados disponibilizados num formato estático, apresentados em diapositivos de PowerPoint. O dinamismo é alcançado com a criação de um pequeno Data Warehouse que permite manter os dados organizados num esquema relacional e na criação de um relatório em Power BI que permite visualizar esses mesmos dados sempre atualizados. Além disso, foi implementado um filtro mensal nas páginas do relatório para apenas observar dados consoante os meses pretendidos. Para

tal, foi necessário desenhar uma arquitetura inicial simples da solução para validar com o cliente, realizar o processo ETL, integração dos dados e por fim construir o relatório.

2.7.1 Arquitetura proposta

A proposta inicial da arquitetura desenhada para a implementação da solução e apresentada e discutida com o cliente pode ser vista na figura 2.6.

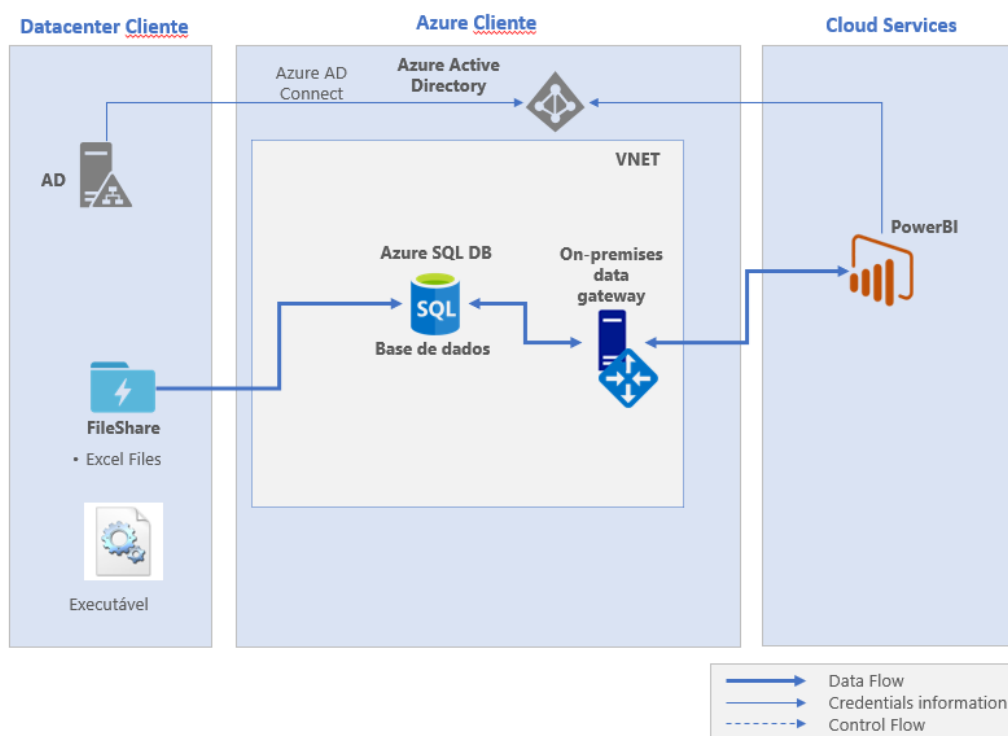


Figura 2.6: Arquitetura da solução inicial

Os dados são disponibilizados em ficheiros Excel e distribuídos por várias tabelas em várias folhas. Associado a estes ficheiros Excel foi criado um ficheiro executável responsável por despoletar, de forma manual, o processo de carregamento destes dados para a base de dados nas infraestruturas Azure do cliente. Após ser feito o tratamento de dados, a base de dados liga-se a um "gateway" de dados no local que aumenta a segurança na fase de transferência entre dados guardados localmente e serviços em Cloud, como por exemplo o Power BI. Por último, com os dados disponíveis no Power BI Desktop, foi possível criar os dashboards pretendidos.

2.7.2 Processo de ETL

Os dados neste projeto foram tratados e modelados dimensionalmente por facilitar a identificação das diferentes entidades no âmbito do projeto. Foi por isso feita uma distinção inicial do que seriam as tabelas de facto e as tabelas de dimensão, da sua relação e os seus atributos. Até ser atingida uma versão final de uma tabela (dimensão ou facto), foi

realizado um processo faseado de criação de várias tabelas, vistas e procedimentos que aumentam a qualidade de tratamento de dados e reduzem a ocorrência de erros.

Inicialmente os dados foram integrados numa tabela que reflete os dados existentes no ficheiro sem qualquer tratamento, sendo que a mesma reside numa base de dados denominada ODS (*operational data store*). Em seguida, foi criada uma vista sobre essa tabela e foram realizadas várias ações de tratamento dos dados, como por exemplo a mudança do nome de atributos, a eliminação de tuplos não pretendidos e a junção de dados que se relacionam. A seguir à configuração da vista, foi criada outra tabela numa outra base de dados - *staging* - que contém todos os atributos da tabela de ODS e uma chave primária como novo atributo. No caso das tabelas de facto, é também adicionado um atributo de chave estrangeira que permite fazer a ligação dos factos com as respetivas dimensões. Após ser feita a criação da tabela em *staging*, foi criada uma vista sobre essa tabela e realizadas pequenas alterações ao modelo de dados. Por último, foi ainda criada mais uma tabela e uma vista numa base de dados DW, sendo que a vista serve como uma versão final do modelo de dados e é importada para a fase seguinte correspondente à criação do relatório.

Como acima referido, foram implementados vários procedimentos que integram este processo. São compostos por duas ações: apagar os dados da tabela que estão guardados no momento anterior à execução do procedimento e atualizá-la com os dados que, do ponto de vista procedimental acima descrito, estão na vista anterior. Desta forma, os dados das várias tabelas criadas estarão sempre atualizados.

2.7.3 Resultados finais e contribuições

Já na ferramenta de construção dos relatórios Power BI, os dados da vista de DW são integrados e ficam disponíveis para serem usados para a criação das visualizações. As diversas funcionalidades da ferramenta permitiram a construção do relatório pretendido, adicionando ainda o filtro temporal e um índice com hiperligações para todas as páginas.

O que foi implementado até esta fase tem diversas vantagens comparando com a solução que o cliente tinha anteriormente, entre as quais a obtenção de um relatório apelativo de dados com a possibilidade de estarem sempre atualizados, uma solução de armazenamento central e permanente guardada na Cloud sempre disponível e com grandes capacidades de armazenamento e a redução significativa de tempo no preenchimento dos dados e criação de relatórios no fim de cada mês.

REQUISITOS E PROPOSTA DE SOLUÇÃO

O cliente iniciou uma fase de transformação interna com o objetivo de dotar o seu vasto grupo empresarial das melhores tecnologias que suportem o seu negócio. Nesse sentido, foi criado um projeto que, entre diversas iniciativas, a mais relevante envolve a implementação de um novo software de suporte ao negócio baseado num ERP (SAP S/4HANA). Um ERP é "um software de gestão de processos de negócio que gere e integra as atividades de finanças, cadeia de fornecimento, operações, relatórios, fabrico e recursos humanos de uma empresa"[30]. Desta forma, ao integrar todos os processos do negócio e os respetivos dados que vão sendo armazenados ao longo do tempo, o mesmo será a principal fonte de informação que irá alimentar o projeto de Business Intelligence que se pretende desenvolver.

Para este projeto, foi identificado um conjunto de necessidades de negócio que procuram ser satisfeitas através de uma solução analítica que incorpore a informação de todas as empresas pertencentes ao cliente, pretendendo-se que seja flexível o suficiente permitindo a integração de novas empresas. Nesse sentido, foram definidas prioridades e lançado um plano com várias fases que têm como objetivo assegurar que as necessidades de consulta e consolidação de dados das diversas empresas do grupo são satisfeitas. Os dados existentes no ERP do cliente são gerados diariamente e são referentes às atividades praticadas pelo mesmo nas várias áreas, englobando conjuntos de dados tais como: informações relativas às vendas de todo o tipo de produções feitas com base em cortiça, desde rolhas até pavimentos e revestimentos, dados sobre as compras efetuadas para a empresa, inúmeros dados sobre os materiais armazenados em stock, informações financeiras e dados sobre o controlo da área responsável pelos transportes.

Nos subcapítulos seguintes será feita uma descrição dos requisitos identificados, da forma com que se pretende abordá-los e distribuí-los em várias fases de implementação da solução analítica, será descrita a arquitetura proposta para os novos modelos responsáveis

por dar resposta aos requisitos identificados e será feita uma descrição mais detalhada sobre o modelo Order to Cash que, no âmbito do projeto, pode ser visto como o modelo central ao mesmo.

3.1 Requisitos de alto nível

Como já descrito no primeiro capítulo, o cliente identificou um conjunto de problemas que levaram à necessidade da construção de uma solução analítica, entre os quais o desajuste entre a informação dos dados disponíveis e as necessidades de análise, a dificuldade de governação e consulta dos dados armazenados e a falta de relatórios flexíveis para consulta por parte dos utilizadores (funcionários da empresa responsáveis pela análise e gestão dos dados das várias áreas). Dessa forma, os requisitos identificados foram distribuídos por quatro blocos principais. (fig.3.1) As principais contribuições desta dissertação incidem sobre os blocos de melhorias sobre os modelos atuais e sobre os novos modelos analíticos.

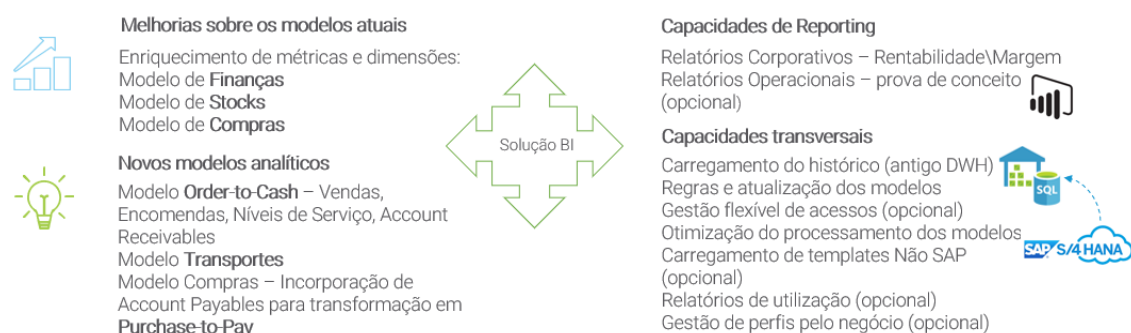


Figura 3.1: Requisitos da solução analítica Imagem presente em documentos internos - Unipartner IT Services)

3.1.1 Melhorias sobre os modelos atuais

Numa primeira abordagem à identificação e organização dos requisitos do projeto, é necessário ter em conta que o cliente já dispõe de uma solução analítica num estado inicial porém inacabado, composta por três modelos tabulares: um modelo de finanças, que integra todos os dados relativos a capital, faturação, custos e margens de venda; um modelo de stocks, responsável pela contabilização das existências, movimentos, coberturas e rotação de stocks e por último um modelo de compras, que armazena informação sobre quantidades, preços, descontos e pesagens. Cada um destes três modelos já tem processos de atualização de dados e os mesmos são amplamente utilizados pelos diversos utilizadores das diferentes unidades de negócio, que, no entanto, identificaram alguns problemas que entram no âmbito deste projeto, no que pode ser visto como uma reestruturação e correção ao que já existe implementado. Por já estarem implementados e a serem utilizados, acresce a importância de perceber como foi implementado, que arquitetura foi montada

e como realizar alterações aos modelos sem que o mesmo implique uma interrupção do funcionamento da solução.

Relativamente aos requisitos de correções e melhoramentos, alguns exemplos seguem-se listados, sendo divididos em requisitos funcionais, como:

- Formatação dos valores;
- Organização de dimensões;
- Enriquecimento de dimensões;
- Implementação de novas dimensões;
- Correção de métricas;
- Implementação de novas métricas;

ou em requisitos não funcionais, como:

- Carregamento de informação via Excel;
- Carregamento do histórico;
- Redução do tempo de processamento do modelo/do volume de dados sem perda de informação.

Para a resolução dos requisitos funcionais e não funcionais, como já acima mencionado, torna-se crucial perceber de que forma o modelo está construído, tanto ao nível da extração e armazenamento dos dados, como ao nível da construção do modelo em si. De forma a não alterar o fluxo normal de funcionamento, pretende-se que quaisquer alterações a serem feitas aos modelos sigam uma lógica semelhante ao que já está montado.

Usando o modelo de finanças como exemplo, é feita uma primeira análise à informação que está a ser extraída e como a mesma é organizada nas várias tabelas de dimensões e de factos. Um facto compreende todos os acontecimentos contabilizados ao longo do tempo, que neste caso corresponde a todos os movimentos contabilísticos e aos valores associados aos mesmos. Por sua vez, as dimensões são as entidades que ajudam a caracterizar e a descrever os factos, possibilitando um maior número de análises que podem ser feitas aos dados. No contexto de finanças, algumas dimensões criadas são o cliente, o tipo de movimento, organização de vendas e a data de lançamento. Desta forma, entre inúmeras análises possíveis, é exequível uma análise dos movimentos filtrando por determinados clientes ou por organizações de venda específicas.

A extração dos dados e consequente armazenamento nas tabelas de dimensões e de factos é feita com base na criação de pacotes de integração que são responsáveis pela extração dos dados da fonte, tratamento e carregamento nas bases de dados já como dimensões ou factos. Cada pacote SSIS criado corresponde a um fluxo de dados, sendo

que em cada fluxo existe uma entrada e uma saída de dados. Sempre que se pretender, os pacotes podem ser executados e os dados percorrem o fluxo desde a origem até ao destino. Na figura 3.2 é possível visualizar os vários pacotes criados para o modelo de finanças com recurso à ferramenta de Integration Services disponível no Visual Studio. A zona (1) apresenta uma lista com o nome de todos os pacotes criados no Integration Services. Quando se pretende analisar o conteúdo de cada pacote, é apenas necessário carregar em cima do pacote pretendido e o fluxo do mesmo aparecerá na zona identificada a (2). Relativamente às conexões criadas, tanto às origens dos dados como ao destino, as mesmas encontram-se identificadas em baixo da zona do fluxo (3).

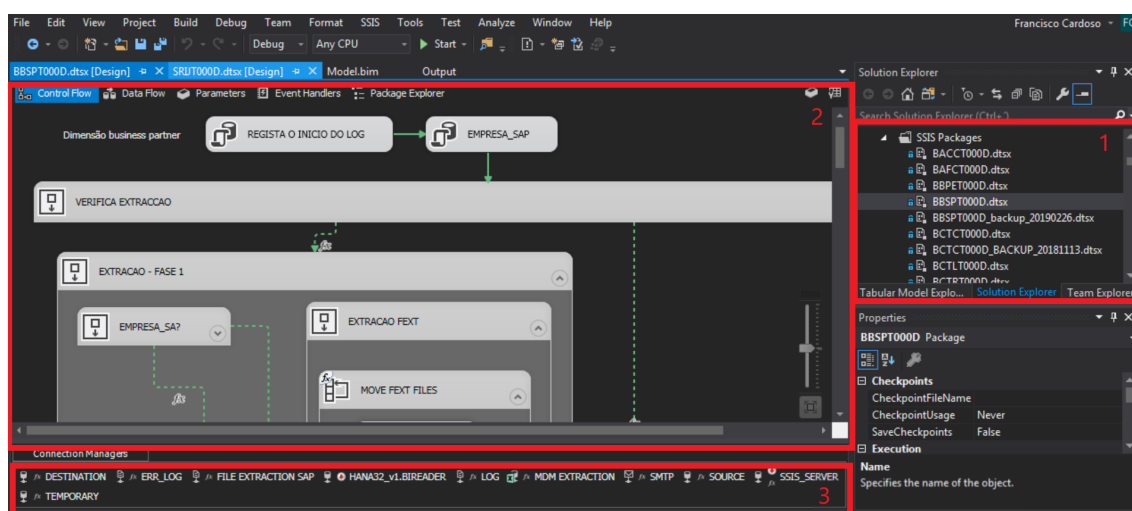


Figura 3.2: Pacotes SSIS do modelo de finanças

Todos os requisitos relativos ao enriquecimento de dimensões ou à criação de novas dimensões irão inicialmente ser endereçadas no Integration Services. No caso de o requisito ser a adição de uma coluna nova a uma tabela de factos ou dimensões existente, terá de ser feito um reconhecimento inicial de quais os pacotes que estão responsáveis por cada tabela. Após ser feito esse reconhecimento, é necessário identificar qual a fonte de onde será feita a extração, seguindo-se o tratamento e adição da nova coluna. No caso de ser necessário adicionar uma nova dimensão ao modelo analítico, será necessário criar um pacote novo responsável por todo o processo ETL. Toda a lógica de implementação dos pacotes SSIS não será alterada nesta fase de desenvolvimento da solução analítica, de forma a não causar possíveis incoerências com o que já está implementado. No entanto, foram utilizadas abordagens diferentes na implementação dos novos modelos analíticos, como no modelo Order to Cash, sendo essas mesmas abordagens descritas nos subcapítulos seguintes.

Para os requisitos relativos à formatação de valores, organização de dimensões/métricas e correção e implementação de novas métricas, os mesmos serão endereçados já no modelo tabular de finanças, que pode ser acedido a partir da ferramenta Analysis Services do Visual Studio. O mesmo pode ser analisado de duas formas distintas: pelo seu

e após serem identificadas todas as métricas que se pretendem atualizar, corrigir ou criar, serão aí feitas todas as alterações.

3.1.1.1 Alterações de implementação

Durante o momento de mapeamento e reconhecimento das implementações já efetuadas no modelos de finanças, stocks e compras, foram identificadas algumas situações relativas ao desenho dos processos de construção que não são consideradas ótimas, como por exemplo na construção dos pacotes SSIS ou pelo facto de existirem vários modelos de dados iguais para cada empresa do cliente. Estes fatores apresentavam claros impactos em termos de tempos de processamento, gestão de anomalias e escalabilidade da solução. Por essa razão, o desenho processual dos novos modelos foi repensado, desde a arquitetura até aos pacotes responsáveis pelo processo ETL, de forma a criar uma solução mais eficiente e estruturada, solucionando todos os requisitos identificados para esta fase. Esta nova proposta de implementação será descrita nos subcapítulos seguintes e aprofundada no capítulo subsequente, sendo apresentadas também as vantagens que se pretendem alcançar com esta nova estrutura.

3.1.2 Implementação de novos modelos

Além de otimizar os modelos já existentes, foi identificada a necessidade de criar novos modelos de forma a cobrir todos os requisitos identificados, entre os quais modelos analíticos que permitam centralizar os dados sobre todas as fases de todas as vendas realizadas, dados sobre todas as fases das compras efetuadas e os dados gerados sobre os transportes. Dessa forma, os novos modelos a implementar são: um modelo Order to Cash, que centraliza todos os dados desde que é feita uma requisição de venda de um cliente até à fase de pagamento por parte do mesmo; um modelo Purchase to Pay, que engloba todos os dados de todas as compras feitas pela empresa, desde a requisição de compra até ao pagamento e um modelo de transportes, que armazena todos os dados relativos ao transporte de mercadorias efetuado pelo cliente.

Nas restantes secções será utilizado o modelo Order to Cash como exemplo descritivo base do tipo de requisitos que se pretendem resolver, qual o intuito do modelo analítico e quais as metodologias que se pretendem utilizar de forma a implementá-lo. A proposta de arquitetura apresentada será transversal aos três novos modelos.

3.1.2.1 Modelo Order to Cash

O modelo de Order to Cash visa disponibilizar aos utilizadores um ponto central, organizado e fiável de consulta a todos os dados referentes às vendas feitas por parte do cliente, desde que é assinado um contrato até ao pagamento por parte dos clientes. Na figura 3.5 é apresentado um diagrama processual composto pelos vários macroprocessos constituintes de um modelo standard Order to Cash.

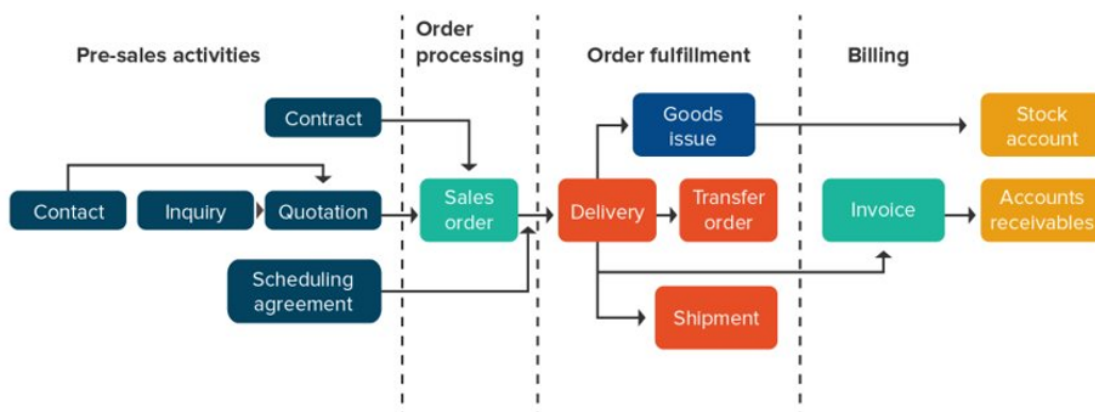


Figura 3.5: Diagrama processual do modelo Order to Cash[31]

No âmbito deste projeto, para as atividades de pré-venda o cliente necessita apenas de analisar os dados referentes aos contratos dado que não utiliza os restantes processos. As várias fases do processo constituintes do fluxo são:

- **Contrato**, que é uma componente de pré-venda e tem como objetivo a formalização das negociações dos termos de venda. O cliente pretende importar para a solução de BI uma visão do módulo de contratos, de modo a medir um conjunto de métricas associadas a esta componente do processo de pré-vendas;
- **Ordem de Venda**, que é o documento que formaliza o compromisso de compra por parte do comprador. O mesmo engloba os produtos a vender, preços e quantidades associadas, quantidades de entrega e as condições de envio e pagamento. O documento pode ter como base o contrato de venda ou pode ser gerado separadamente;
- **Remessa**, que corresponde ao ato de fazer seguir uma encomenda de acordo com o estipulado na ordem de venda. Cada remessa tem uma quantidade associada e cada ordem de venda pode ter várias remessas associadas. O cliente pretende dispor na solução de BI de vários indicadores que permitam analisar os dados relativos às várias expedições realizadas;
- **Fatura**, que corresponde ao documento que efetua o lançamento nas contas relacionadas do comprador no sistema contabilístico, habitualmente com base na remessa;
- **Tesouraria**, que é a fase do processo que engloba as atividades de pagamento, tais como o débito na receita e a creditação no comprador.

De forma a construir o modelo analítico de um modo faseado e estruturado, foi utilizado o diagrama processual apresentado anteriormente como base do desenho do modelo, visto que o mesmo apresenta os blocos principais de cada fase do processo. De um ponto

de vista mais técnico, o diagrama também permite ter uma visão em como os dados serão armazenados no modelo analítico, sendo possível identificar à partida quais as tabelas que irão armazenar factos e as que irão armazenar dimensões. Por exemplo, no caso do bloco das faturas (apresentado como Invoice na imagem), o mesmo representa todas as faturas criadas e armazenadas em sistema, cada uma associada a uma venda efetuada pelo cliente. Nesse sentido, cada fatura irá corresponder a um facto, i.e., um acontecimento no tempo e no contexto do negócio que contém informação que pode ser medida. As faturas serão então guardadas numa tabela de factos na base de dados e cada fatura, que corresponde a uma linha da tabela, terá ligações com outras tabelas - as tabelas de dimensões - que irão permitir filtrar e analisar as faturas com mais detalhe (3.6). Algumas dimensões de exemplo são o *Business Partner*, Organização de Vendas, Centro de Custo e o Material.

(C) 2008 datawarehouse4u.info

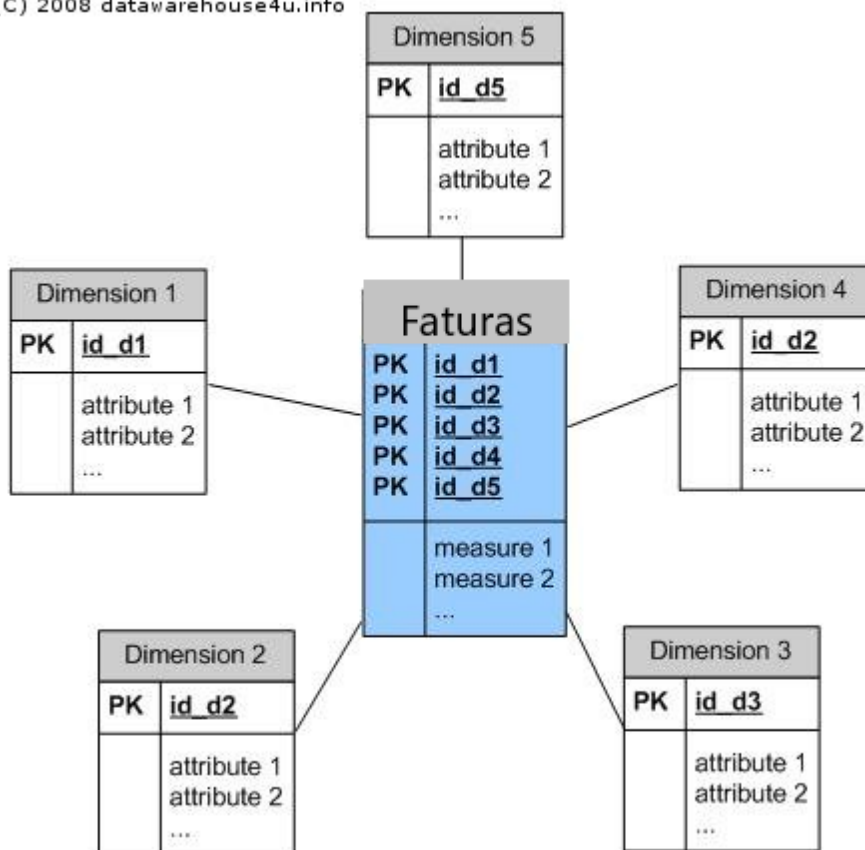


Figura 3.6: Desenho do esquema em estrela do bloco de faturas (modificado) [32]

Após serem identificadas as várias tabelas que se pretendem criar e quais as ligações entre tabelas que se pretendem construir, o principal objetivo a partir deste ponto é o de realizar o processo de ETL de forma a extrair os dados da fonte, tratar dos mesmos e inseri-los na base de dados que servirá de fonte ao modelo analítico. O processo a ser construído terá de representar um fluxo de dados contínuo que permita que seja executado em qualquer momento. A partir do momento em que os dados ficam disponíveis no modelo,

o mesmo pode ser consultado livremente por diversas vias, como por exemplo a partir de pivot tables em Excel, relatórios criados em Power BI ou criados no Reporting Services ou apenas consultados ad-hoc a partir do modelo, apresentando para todos os casos toda a informação atualizada recolhida ao longo do tempo por parte da empresa e dessa forma auxiliando nas tomadas de decisão sobre o negócio.

Além de se disponibilizar a consulta livre dos dados, existe também a possibilidade de se criarem métricas sobre o modelo analítico, fazendo parte dos requisitos principais identificados pelo cliente para este projeto. O objetivo destas métricas é o de agregar e pré-calcular dados de forma a devolver informação frequentemente consultada de forma rápida e sempre atualizada. Existem vários indicadores já identificados pelo cliente, nomeadamente indicadores relativos a níveis de serviço nas várias fases do processo, como por exemplo no momento da ordem de venda em que se pretende analisar o tempo de registo ou o tempo de planeamento e no momento da remessa, em que se pretende construir indicadores que calculem, por exemplo, o intervalo de tempo decorrido entre a data do pedido do comprador e a data desejada de saída da mercadoria. Durante a criação das tabelas que dão suporte ao modelo analítico, é preciso ter em conta se as mesmas conseguem satisfazer todos os indicadores já identificados pelo cliente.

3.1.2.2 Arquitetura proposta

A solução que se pretende construir para cada um dos novos modelos pode ser dividida em três camadas principais (3.7): Uma camada onde se centram as fontes de dados, uma camada onde será tratada e mantida a informação e uma camada que representa o consumo por parte dos utilizadores dessa mesma informação.



Figura 3.7: Arquitetura proposta para os novos modelos (Imagem presente em documentos internos - Unipartner IT Services)

Relativamente à camada da fonte de dados, a mesma irá ser composta por várias fontes. Além da principal já anteriormente mencionada - o ERP SAP - espera-se também que seja possível integrar dados de empresas que não estão a ser geridas no ERP SAP, pelo que a sua informação será carregada a partir de folhas Excel. Além disso e como será posteriormente esclarecido, existe informação adicional que permite enriquecer algumas tabelas de dados já existentes e que é mantida paralelamente em Sharepoint, pelo que essa informação também terá de entrar no fluxo de dados a ser criado.

Quanto às bases de dados que constituem o DW onde os dados serão mantidos, pretende-se que o mesmo seja criado com recurso a Azure SQL DB's. Esta opção justifica-se por três razões principais: a primeira deve-se ao facto das bases de dados serem mantidas na plataforma de computação em Cloud Windows Azure, o que permite beneficiar de vantagens tal como a alta performance de armazenamento, a elevada segurança dos dados e a rápida escalabilidade do DW de acordo com as necessidades do cliente. A segunda razão deve-se ao facto das bases de dados Azure se basearem em tecnologias do SQL Server que, para o desenvolvimento do projeto, é a ferramenta utilizada para criar e modificar todos os objetos das bases de dados, i.e. todas as tabelas, vistas e procedimentos. Por último, a escolha das componentes da arquitetura irá incidir preferencialmente em ferramentas Microsoft pelo facto de, como já anteriormente mencionado, a Unipartner ser um parceiro estratégico da Microsoft e o uso das suas ferramentas trazer benefícios.

Por fim, a camada de visualização engloba todas as componentes que permitem aos utilizadores do negócio acederem aos modelos analíticos implementados, sendo que esse acesso pode ser feito por diferentes vias e o mesmo corresponde à versão mais atual dos dados.

Quanto ao processo ETL a ser realizado, o mesmo é composto por várias fases e tem o objetivo de fazer chegar os dados desde a sua fonte até à versão final no respetivo modelo analítico. De forma a garantir uma boa esquematização e qualidade dos dados que vão ser usados nas camadas de visualização, os mesmos passam por um processo sequencial de tratamento que envolve a criação de algumas bases dos dados relacionais (3.8). Para o efeito, pretendem-se criar pacotes SSIS responsáveis pela extração, tratamento e carregamento dos dados, sendo os mesmos responsáveis pelo funcionamento do fluxo.

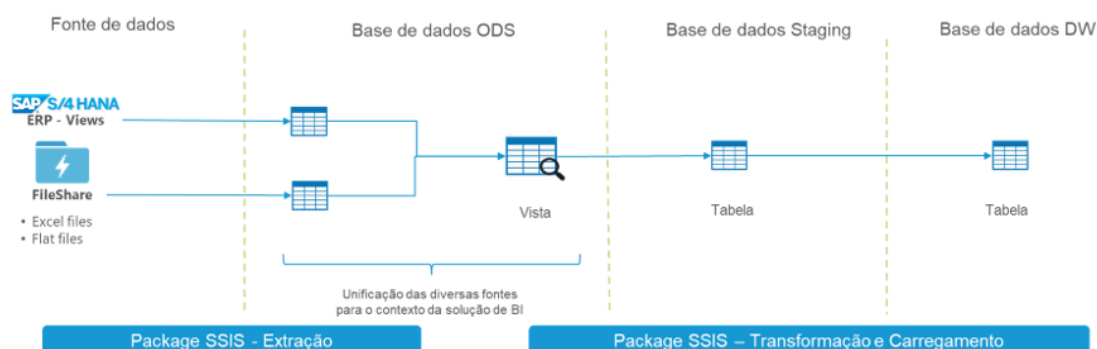


Figura 3.8: Esquema proposto do processo ETL (Imagem presente em documentos internos - Unipartner IT Services)

A primeira base de dados a ser criada e populada denomina-se de ODS (*Operational Data Store*). O objetivo desta base de dados é o de criar uma imagem exata dos dados tal e qual como se encontram nos sistemas operacionais. Esta decisão traz algumas vantagens, tais como a de ter nas infraestruturas do cliente uma versão exata dos dados que se encontram na fonte e por isso ser mais fácil de manipulá-los a partir desse ponto, ao

invés de manipular os dados no mesmo momento em que se faz a extração dos mesmos. Isto permite estruturar da melhor forma o processo ETL e auxilia tanto no processo de correção de eventuais erros, como por exemplo campos não preenchidos ou campos ambíguos, como na normalização e otimização do tipo de dados. A partir da base de dados ODS agora composta por inúmeras tabelas provenientes das fontes, são criadas várias vistas ODS que irão corresponder às futuras tabelas de factos e dimensões do modelo analítico. Todas as transformações e modelações principais aos dados que provêm das tabelas ODS serão realizadas e refletidas nas vistas ODS. A partir das vistas ODS, serão criados procedimentos responsáveis pela passagem dos dados das vistas ODS para as tabelas de staging na base de dados de staging e desta para as tabelas da base de dados de DW, que correspondem à versão final das tabelas que são a fonte do modelo analítico. Podem ocorrer pequenas alterações e transformações aos dados entre as vistas ODS e tabelas de staging e entre staging e as tabelas de DW. No entanto, pretende-se seguir sempre duas regras de implementação principais: que as alterações principais aos dados estejam já refletidas nas vistas de ODS e que as tabelas presentes no Data Warehouse contenham sempre a versão mais recente e otimizada dos dados.

No capítulo seguinte será apresentado com mais detalhe a forma de implementação das várias tabelas que suportam o modelo tabular Order to Cash. Será ainda descrito um exemplo de criação do processo ETL, dos pacotes SSIS responsáveis pelo funcionamento do fluxo, das vistas criadas e dos procedimentos implementados, além de ser explicado como é importada a informação da base de dados de DW e como a mesma permite a criação dos indicadores requisitados.

IMPLEMENTAÇÃO

Neste capítulo será exposto de que forma foi implementado o modelo Order to Cash, dando vários exemplos concretos das fases de implementação e evidenciando onde cada uma encaixa no fluxo geral. A razão de se apresentar apenas um modelo deve-se ao facto de que, logicamente, os novos modelos são semelhantes, pelo que a exposição de um modelo é suficiente à compreensão dos outros dois. Por fim, será mostrado um *overview* do resultado final alcançado e será discutido no subcapítulo de avaliações se o mesmo satisfaz os objetivos estabelecidos inicialmente.

Para os novos modelos foi desenhada uma nova estrutura comparativamente à usada pelos modelos já existentes. Além disso, a lógica de implementação dos pacotes SSIS também foi redesenhada e será explicado todo o processo de implementação desde a identificação dos dados que se pretendem extrair até à criação do modelo tabular. Pretende-se também justificar ao longo dos subcapítulos algumas decisões de implementação e as vantagens que se obtêm de tais escolhas.

Relativamente às tarefas realizadas que contribuíram para as alterações aos modelos já existentes, outras semelhantes estarão disseminadas na implementação do novo modelo, tornando perceptível o tipo e a complexidade das mesmas.

4.1 Arquitetura dos novos modelos

A arquitetura construída para os novos modelos segue a esquematização proposta no capítulo anterior, sendo subdividida em três camadas: camada onde residem as fontes de dados, a camada de armazenamento dos dados onde se encontra o DW e a camada de visualização onde os dados são consumidos pelos utilizadores.

A camada de fontes é composta por quatro componentes:

- ERP SAP - corresponde à fonte principal de dados, armazenando quase toda a

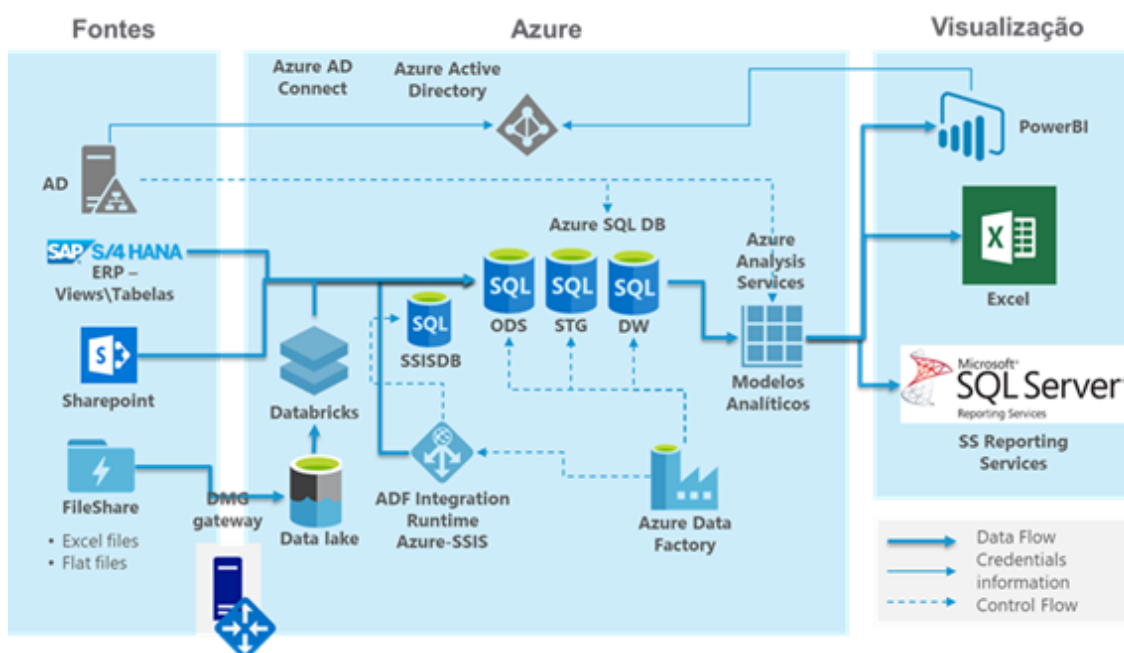


Figura 4.1: Arquitetura dos novos modelos (Imagem presente em documentos internos - Unipartner IT Services)

informação criada diariamente pelo cliente;

- Sharepoint - contém informação adicionada manualmente que enriquece as tabelas principais;
- FileShare - é composto por vários ficheiros Excel que correspondem ao carregamento de informação das empresas do cliente que não estão a ser geridas em SAP;
- Active Directory - responsável pela gestão de permissões de acesso às três componentes apresentadas anteriormente e às ligações efetuadas entre as fontes de dados e o DW.

Os dados provenientes da camada de fontes passam pelo processo de ETL e são armazenados e mantidos na segunda camada, a camada Azure, composta pelas seguintes componentes:

- Um Azure Data Lake - que disponibiliza, do ponto de vista da implementação, os recursos necessários a todo o tratamento feito aos dados;
- Uma componente de Databricks responsável pelo processamento em larga escala de todos os dados;
- Uma componente Azure Data Factory que realiza o processamento diário dos dados para a solução de BI;

- Três Azure SQL DB's - ODS, Staging e DW - que contêm todos os objetos que armazenam os dados para cada fase do processo ETL, sendo os mesmos tabelas, vistas e procedimentos;
- Uma componente que corresponde ao modelo analítico criado em SQL Analysis Services.

Após ser construído o modelo analítico e ser feito o *deploy* ao mesmo, o modelo pode ser acedido por diversas formas que constituem a terceira camada da arquitetura - a camada de visualização. No âmbito do projeto, foram identificadas três formas distintas de acesso aos modelos:

- Excel, sendo a forma mais rápida de se criarem pivot tables sobre o modelo analítico, que permitem aos utilizadores criar tabelas de análise aos dados e cruzá-las com as dimensões pretendidas;
- SQL Reporting Services, uma ferramenta integrada no Visual Studio que permite criar relatórios dinâmicos;
- PowerBI, que não se encontra no âmbito inicial do projeto mas que é vista como a ferramenta mais vasta e completa à criação de relatórios sobre os modelos de dados.

4.1.1 Vantagens

Existem inúmeras vantagens de se estabelecer uma arquitetura por camadas, em que todas se encontram interligadas entre si e estão bem definidas quanto ao seu âmbito.

Uma das vantagens é o facto de reduzir a complexidade de implementação. Estando cada camada bem definida quanto ao seu objetivo, todas as componentes que são adicionadas à arquitetura têm à partida uma camada associada, que como já referido comunica eficientemente com as restantes camadas. Esta separação bem definida das várias componentes ajuda à compreensão geral da arquitetura e promove uma gestão fácil da mesma. Além disso, componentes que necessitam de uma comunicação constante são integradas na mesma camada, o que aumenta a performance de execução das tarefas. Outra vantagem que se destaca é o facto de auxiliar na escalabilidade da solução. Podendo sofrer várias mudanças ao longo do tempo, pretende-se que a solução de BI implementada permita a fácil integração de novas componentes. Por ser criada esta separação por camadas, à partida a integração de uma componente nova apenas tem de ser sincronizada dentro da sua camada, visto as ligações feitas às restantes já estarem previamente estabelecidas. A especialização de cada componente também reduz o custo associado ao aumento das capacidades em áreas funcionais específicas, pois permite identificar as camadas que necessitam de um aumento de recursos e centrar as alterações nessas camadas sem obrigar a evoluir toda a solução.

4.2 Modelo Order to Cash

Como descrito no capítulo anterior, o modelo Order to Cash é composto por várias fases, sendo as mesmas a fase de contratos, ordens de venda, remessas, faturas e a fase de tesouraria. Em cada uma das fases, existe um conjunto de dados que vai sendo gerado e armazenado nas infraestruturas do cliente e que, ao ser tratado, permite retirar informações relevantes no âmbito do negócio. Seguindo a arquitetura apresentada para os novos modelos, de seguida é apresentado o processo dos dados desde a sua fonte até ao modelo analítico, modelo esse que já permite aos utilizadores consultar a informação pretendida. Sendo o processo muito semelhante entre as várias fases, apenas será apresentado o fluxo relativo à fase de faturas, ou seja, todos os dados relativos à faturação armazenada pelo cliente.

4.2.1 Identificação e mapeamento dos dados

O primeiro passo quando se pretende realizar a extração de uma fonte de dados é o de identificar claramente quais os dados que se pretendem ter na solução final de BI. Por vezes essa identificação é fácil, visto os dados na fonte terem eventualmente o mesmo nome na solução final ou simplesmente porque a extração será feita a todos os dados que existem guardados. No entanto, neste caso os nomes das várias colunas e tabelas que residem na fonte têm nomes próprios do sistema em que foram criadas, o que torna necessário realizar um passo adicional de mapeamento. Além disso, é crucial que se identifique as tabelas e colunas que se pretendem extrair, pois uma extração total dos dados da fonte significaria um aumento do tempo e complexidade considerável no processo diário dos dados e no volume presente no DW. Uma das formas de identificação da informação necessária a ser extraída é com base nos indicadores que se pretendem construir no modelo.

Na figura 4.2 é possível observar uma parte de uma matriz de análise aplicada à fase da faturação.

No eixo horizontal foram dispostos todos os indicadores previamente identificados pelo cliente. Em seguida, foram identificados para cada indicador todas as dimensões que permitem realizar uma análise relevante sobre esse indicador. Por exemplo, o indicador *Nr. de Faturas* devolve o número total de faturas criadas até ao momento. No entanto, será também relevante calcular o número de faturas num período temporal específico, ou o número de faturas que contenham um certo tipo de material. Dessa forma, na solução analítica terá de constar uma dimensão 'Data' e outra 'Material', de forma a poder ser feito o cruzamento com a factual de faturas. Após ser feita esta identificação para cada indicador, o resultado final é uma matriz como a apresentada na figura 4.2, em que as dimensões aparecem listadas no eixo vertical e os cruzamentos devidamente identificados com os indicadores.

Após terem sido identificadas todas as dimensões que se pretendem trazer para o

	Nr. de Faturas	Nr. de linhas por fatura	Nr. de Documentos Associados à Fatura	Margem de Venda Faturado	Vendas faturadas	Faturas sem Documento Contábil
Tipo Fatura	x	x	x	x	x	
Nr. do documento	x	x	x	x	x	
Data de facturamento	x		x	x	x	
Status do lançamento	x		x	x	x	x
Condição de pagamento	x			x	x	
Empresa	x			x	x	
Condições de Pricing	x			x	x	
Material	x			x	x	
Hierarquia Material	x			x	x	
Business Partner	x			x	x	
Centro	x			x	x	
Centro de Lucro	x			x	x	
Hierarquia Centro de Lucro	x			x	x	
Centro de Custo	x			x	x	
Hierarquia Centro de Custo	x			x	x	
Área de contabilidade de custos	x			x	x	
Local de expedição	x			x	x	
Canal de distribuição	x			x	x	
Setor de atividades	x			x	x	
Depósito	x			x	x	
Região de Venda	x			x	x	
Organização de Vendas	x			x	x	
Canal de Distribuição	x			x	x	

Figura 4.2: Matriz de análise

modelo, é necessário realizar a fase de mapeamento. Na fonte SAP, a tabela que contém toda a informação sobre cada material e que irá originar na solução de BI uma dimensão com o mesmo nome, não se encontra assim denominada. O mesmo acontece para as restantes tabelas de dimensões e de factos. Para este projeto, grande parte do mapeamento foi realizado por uma empresa parceira com competências na área. De forma a simplificar o processo de mapeamento e posterior implementação, foi criado um ficheiro Excel em que cada página corresponde a uma dimensão, tendo cada uma o nome da tabela ou tabelas que contém a informação necessária para a construção dessa mesma dimensão. Além do nome das tabelas, foram também mapeadas apenas as colunas que se pretendem extrair (fig. 4.3).

Após ser feito o mapeamento das tabelas e colunas que se pretendem extrair, é possível iniciar o processo de extração dos dados, criando os respetivos pacotes e dando assim início ao processo de ETL. A título de exemplo de um processo completo implementado, será usada a dimensão Material, em que o mapeamento completo dos campos já se encontra na figura 4.3.

Fonte			Dimensão
Sistema	Tabela/Lista	Coluna	Coluna
SAP	SAPABAP1.MARA	MATNR	[CodMaterial]
SAP	SAPABAP1.MARA	MATKL	[CodGrupoMercadoria]
SAP	SAPABAP1.MARA	PRDHA	[CodHierarquia]
SAP	SAPABAP1.MARA	MTART	[CodTipoMaterial]
SAP	SAPABAP1.MAKT	MAKTX	[DesMaterialPT]
SAP	SAPABAP1.MAKT	MAKTX	[DesMaterialEN]
SAP	SAPABAP1.T023T	WGBEZ60	[DesGrupoMercadorias]
SAP	SAPABAP1.T023T	WGBEZ	[DenominacaoGrupoMercadorias]
SAP	SAPABAP1.T134T	mtbez	[DesTipoMaterial]
SAP	SAPABAP1.MARA	MEINS	[UnidadeMedidaBasica]
SAP	SAPABAP1.ZMM_MAT_CM_T	MATNR_CM	[MaterialDestino]
SAP	SAPABAP1.ZMM_REVAL_PRIC_T	BUKRS	[PrecoRevalorizado]
SAP	SAPABAP1.T179T	VTEXT	[Nivel1PT]
SAP	SAPABAP1.T179T	VTEXT	[Nivel2PT]
SAP	SAPABAP1.T179T	VTEXT	[Nivel3PT]
SAP	SAPABAP1.T179T	VTEXT	[Nivel4PT]
SAP	SAPABAP1.T179T	VTEXT	[Nivel5PT]
SAP	SAPABAP1.T179T	VTEXT	[Nivel1EN]
SAP	SAPABAP1.T179T	VTEXT	[Nivel2EN]
SAP	SAPABAP1.T179T	VTEXT	[Nivel3EN]
SAP	SAPABAP1.T179T	VTEXT	[Nivel4EN]
SAP	SAPABAP1.T179T	VTEXT	[Nivel5EN]

Figura 4.3: Mapeamento dos dados da dimensão Material

4.2.2 Extração

De forma a facilitar a identificação e gestão dos vários pacotes que vão sendo criados no Visual Studio, decidiu-se estabelecer algumas regras de nomenclatura. A mesma define a nomeação dos pacotes como a concatenação do nome da tabela principal da fonte de dados, seguido de E, T ou L no caso de ser um pacote de extração, transformação ou carregamento, respetivamente, e por último o nome da tabela que contém a informação para a construção da dimensão. Na figura 4.4 é possível observar o fluxo geral do pacote de extração MARA - SAPABAP1_E_MARA - que contém as colunas que armazenam todos os códigos dos materiais, além dos códigos do grupo de mercadorias, da hierarquia do material e do tipo de material.

Este pacote é formado por um fluxo de controlo que tem dois objetivos principais:

- Eliminar todos os dados existentes na tabela da base de dados de ODS, previamente criada à imagem da estrutura presente na fonte (apenas as colunas que se pretendem);
- Inserir novamente os dados mais recentes da fonte na tabela.

Para tal, é utilizado um bloco de execução "Execute SQL Task" que irá fazer o *truncate* à tabela ODS. A task recebe duas variáveis de forma a identificar devidamente a tabela que se pretende "limpar", uma com o nome do esquema e outra com o nome da tabela. Após ser executada a task, segue-se um bloco "Data Flow Task" que contém um fluxo de dados responsável pela passagem dos dados da fonte para a tabela MARA na base de dados ODS, agora vazia, finalizando assim o processo de atualização dos dados.

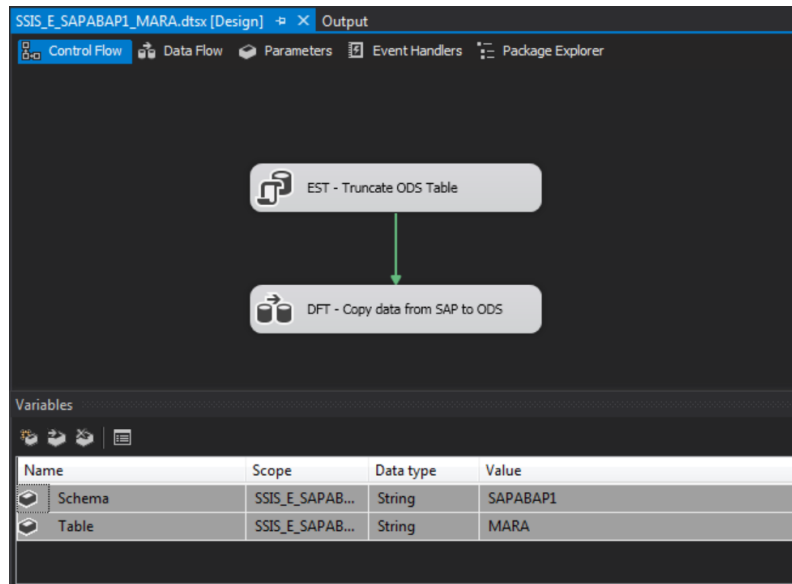


Figura 4.4: Pacote de extração MARA

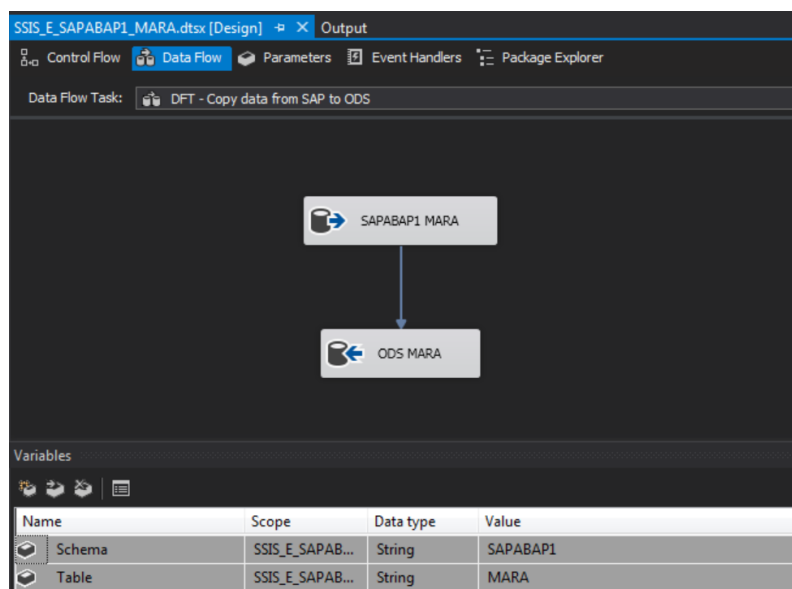


Figura 4.5: Fluxo de dados da passagem de dados da fonte para ODS

No fluxo, o primeiro bloco é responsável pela seleção de todas as colunas pretendidas da fonte, fazendo em seguida a passagem dos dados selecionados para o segundo bloco. O segundo bloco é responsável por armazenar os dados extraídos, mapeando as colunas da fonte com as da tabela em ODS. Esse processo é facilitado pelo facto da tabela em ODS ter a mesma estrutura da tabela em SAP. Concluída a execução do pacote, os dados estão prontos para serem manipulados e transformados.

4.2.2.1 Criação da vista ODS

Após a criação da tabela ODS e a mesma ser constituída por todas as colunas extraídas da tabela fonte necessárias à construção da dimensão, segue-se a criação da vista ODS. A criação da vista ODS incide já em parte na fase de transformação, visto aglomerar o maior número de alterações a serem feitas de forma a ser criada uma ideia inicial de como será materializada a dimensão. Algumas dessas alterações incluem a eliminação de tuplos repetidos, a nomeação correta dos campos, a junção de colunas de tabelas diferentes e a aplicação de algumas condições. A decisão de se criar uma vista nesta fase do processo deve-se a três razões principais:

- Permite manipular facilmente várias colunas e tabelas, criar condições e eliminar linhas não pretendidas, sendo que essas alterações se traduzem numa tabela simplificada à imagem do que se pretende. Quaisquer alterações que sejam necessárias realizar à estrutura dessa mesma tabela centralizam-se na *query* criada que origina a vista;
- Oferece uma camada de proteção adicional dos dados, visto que a criação das vistas pode restringir as colunas ou tabelas que se pretende que sejam acedidas pelos utilizadores finais;
- Por apresentar os dados de forma consistente e à semelhança da estrutura final, serve como fonte de dados ao resto do processo, i.e, é sempre criada uma vista no final de cada pacote (E,T,L) e a mesma serve como a fonte de dados que será utilizada pela fase seguinte no processo. No caso do pacote de carregamento, a vista criada sobre a tabela em DW será o ponto de ligação com o modelo tabular.

Para o caso específico da vista ODS da dimensão Material, foram realizadas várias junções com outras tabelas extraídas da fonte de forma a adicionar colunas relevantes, como colunas que contêm descrições que complementam os códigos existentes. Adicionalmente, foi criada uma hierarquia descritiva dos materiais com cinco níveis, disponibilizando, tal como para o resto das descrições, uma coluna em português e uma em inglês. Esta necessidade deve-se ao facto de existirem vários utilizadores que não têm a língua portuguesa como língua materna. Relativamente à criação de uma hierarquia, a mesma

traz benefícios pois permite uma maior organização de dados que contenham informações semelhantes mas que apresentam níveis de especificidade diferentes. A criação da vista pode ser consultada no anexo I.

4.2.3 Transformação

Na fase de transformação, foi também criado um pacote SSIS que executa um processo de atualização semelhante ao implementado anteriormente, sendo que desta vez incide sobre a base de dados *staging*. Os objetivos são o de:

- Realizar o *truncate* à tabela de staging;
- Fazer a passagem dos dados de ODS para a tabela em staging, preparando-os para serem integrados no DW.

Na fase de transformação pretende-se finalizar a transformação feita aos dados, sendo que a maior parte já foi realizada na criação da vista ODS. Foi apenas identificada uma alteração final, transversal a todas as dimensões e factuais criadas - a criação de uma chave primária. A mesma foi criada recorrendo a uma atribuição numérica sequencial automática a todas as linhas da tabela, como um índice, contribuindo de forma rápida para integridade dos dados. A presença de uma chave primária nas dimensões é fundamental pois identifica univocamente cada linha da tabela e permite realizar as conexões necessárias com as factuais.

4.2.4 Carregamento

Após terminado o processo de transformação dos dados, o último passo é o de criar as vistas que correspondem à versão final da estrutura de dados final presente no DW. Essas vistas serão o ponto de ligação com o modelo analítico para onde os dados serão carregados.

O pacote responsável pelo carregamento dos dados tem a mesma lógica dos pacotes apresentados anteriormente, tendo sido acrescentado um passo adicional de forma a contornar um problema evidenciado. O problema surge pelo facto do modelo analítico não permitir estabelecer conexões com tabelas em EDW que contenham conexões com outras bases de dados, neste caso de staging. Dessa forma, decidiu-se criar uma tabela auxiliar na base de dados em EDW (*enterprise data warehouse*) que apenas serve como ponte para a tabela final. Os dados são posteriormente processados da tabela auxiliar para a tabela final com recurso a um procedimento que, de forma a tornar o processo eficiente, apenas copia as linhas novas que ainda não estão presentes na tabela final fig. 4.6. No anexo I encontra-se o comando de criação do procedimento.

4.2.5 Modelo tabular

Estando o processo de ETL concluído, a última etapa da implementação corresponde à criação do modelo tabular. Com recurso à ferramenta SQL Analysis Services, pretende-se:

- Identificar todas as vistas em EDW de dimensões e factos que se pretendem importar para o modelo e importá-las;
- Fazer a ligação entre as chaves estrangeiras das factuais com as chaves primárias das dimensões;
- Esconder todas as colunas que não são necessárias para as análises realizadas pelos utilizadores, como colunas que guardam códigos ou que foram criadas para auxiliar à implementação de outras;
- Renomear as colunas de acordo com as intenções do cliente;
- Processar as tabelas no modelo para refletirem os dados presentes na base de dados de EDW.

Após serem executados estes passos, a estrutura do modelo tabular fica finalizada. O diagrama de uma perspectiva criada sobre o modelo pode ser consultado na figura 4.8.

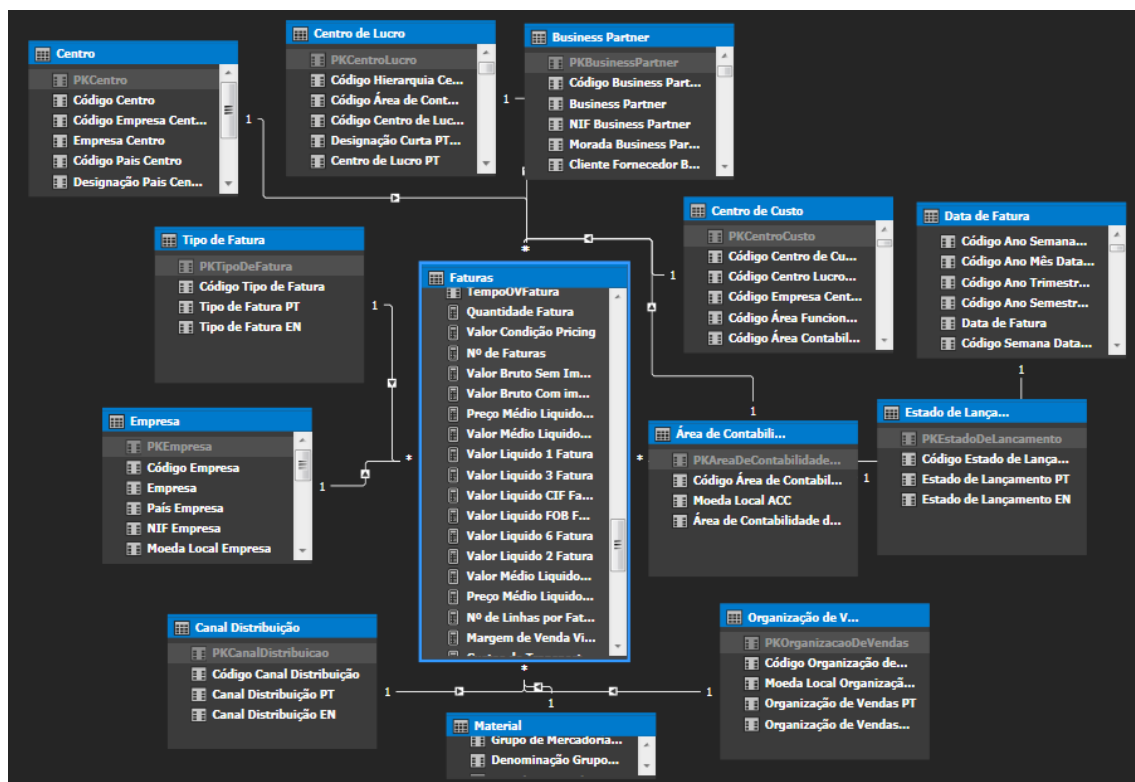


Figura 4.8: Diagrama da perspectiva de faturas

Uma perspectiva permite criar uma visualização filtrada do modelo completo, de forma a disponibilizar aos utilizadores das várias áreas de negócio apenas as tabelas, colunas e

métricas de que necessitam. Na figura, é possível observar, ao centro, a factual de faturas ligada às várias dimensões de análise: *Business Partner*, Centro de Lucro, Centro de Custo, Data da Fatura, Área de Contabilidade de Custos, Estado de Lançamento, Organização de Vendas, Canal de Distribuição, Empresa, Tipo de Fatura, Centro e a dimensão que serviu de exemplo à demonstração do processos de implementação - Material.

4.3 Resultado e avaliação

Findo o processo de importação de todas as tabelas que constituem o modelo Order to Cash e feitas as respetivas ligações e renomeações das colunas, o resultado final do modelo tabular irá conter todas as factuais que representam cada um dos blocos do fluxo apresentado no capítulo 3, ou seja, uma factual que armazena os contratos, outra para as ordens de venda, outra referente às guias de remessa e uma factual de faturas. Além disso, o modelo também irá conter todas as dimensões que permitem realizar todas as análises requeridas pelo cliente.

Relativamente à avaliação da solução implementada, a mesma é efetuada pelos vários utilizadores do negócio do cliente que validam se os dados presentes na solução estão em conformidade com os dados presentes no ERP. Uma das formas de realizar esta validação é comparando os resultados dos indicadores implementados na solução analítica com o mesmo indicador calculado manualmente, dado que um mesmo resultado significa que não houve adulteração dos dados durante o processo de construção do modelo e que a fórmula de cálculo do indicador está correta. Para a maioria dos indicadores implementados, o resultado obtido pela implementação manual foi facultado, pelo que a validação pôde ser efetuada durante a implementação.

A mesma forma de validação também ocorreu para os novos indicadores criados nos modelos da primeira fase - finanças, stocks e compras. Um exemplo de construção e validação de um indicador é apresentado de seguida.

O indicador "Mix Produção Ordem", criado no modelo de Stocks, é dado pela divisão do valor da produção sobre a quantidade da produção. O mesmo é válido para todos os centros e considera-se apenas os movimentos do tipo 'AJUSTES' e 'PRODUÇÃO'.

A fórmula de cálculo criada manualmente foi facultada numa página Excel, em que foram adicionalmente aplicados filtros sobre o cálculo do indicador de forma a dar um exemplo prático de utilização. Os filtros indicam que o cálculo do "Mix Produção Ordem" é apenas para o mês de Dezembro do ano de 2018, para o centro com o código 1104 e em que o grupo de mercadorias tenha o código 5001 e o tipo de produto seja ZV50.

Por sua vez, o indicador foi criado no modelo tabular recorrendo à linguagem DAX que, como já referido, é uma linguagem de manipulação de dados que potencia e acelera as análises feitas aos dados. A fórmula do mesmo pode ser consultada em baixo.

```
1  
2 Mix Producao_Ordem :=
```

```

3
4  VAR Valor =
5      CALCULATE('Stocks - Movimentos'[Movimentos - Valor]; 'Tipo de Movimento
de Stock'[Grupo1 Tipo de Movimento de Stock] IN {"Ajustes"})
6      +CALCULATE('Stocks - Movimentos'[Movimentos - Valor]; 'Tipo de
Movimento de Stock'[Grupo1 Tipo de Movimento de Stock] IN {"Producao"})
7
8  Var Quantidade =
9      CALCULATE('Stocks - Movimentos'[Movimentos - Quantidade Base]; 'Tipo de
Movimento de Stock'[Grupo1 Tipo de Movimento de Stock] IN {"Ajustes"})+
10     CALCULATE('Stocks - Movimentos'[Movimentos - Quantidade Base]; 'Tipo de
Movimento de Stock'[Grupo1 Tipo de Movimento de Stock] IN {"Produç o"})
11
12  Return DIVIDE(Valor;Quantidade)
13
14
15
16

```

Na mesma é possível observar a criação de duas variáveis, uma para o valor e outra para a quantidade, em cada uma delas considera apenas os tipos de movimento pretendidos. Por fim, é realizada a divisão do valor pela quantidade e a fórmula é armazenada no modelo tabular.

Aplicando os mesmos filtros descritos anteriormente à fórmula, confirma-se que os valores coincidem e o indicador fica validado. Para uma validação adicional, alteram-se os filtros em ambos os cálculos e procede-se a uma nova validação de igualdade dos valores.

O mesmo processo repetiu-se para os vários indicadores criados nos modelos de finanças, stocks, compras e Order to Cash.

CONCLUSÕES

Nesta dissertação pretendeu-se solucionar os problemas evidenciados por um cliente que apresentava dificuldades em organizar e extrair informação útil dos seus dados armazenados, de forma a auxiliar no processo de tomadas de decisão para o negócio. Para tal, foi desenvolvido um projeto de Business Intelligence que resultou no melhoramento de acesso aos dados com criação de vários modelos analíticos. Os mesmos respondem às necessidades identificadas pelo cliente, pois apresentam os dados intuitivamente organizados e disponíveis para serem analisados de inúmeras formas, além de serem criados vários indicadores sobre estes dados que diminuem o tempo necessário para a consulta das análises mais recorrentes. O projeto dividiu-se em quatro fases, sendo que apenas as primeiras duas entraram no âmbito das atividades realizadas para esta dissertação: inicialmente foram feitas as melhorias aos modelos já existentes na solução antiga do cliente e numa segunda fase foi implementado um novo modelo de raiz, o modelo Order to Cash, sendo proposta uma nova arquitetura e metodologia de implementação comparativamente à solução existente. Esta nova arquitetura trouxe vários benefícios, entre os quais a diminuição da complexidade, o aumento de performance e o aumento da escalabilidade da solução.

Os resultados obtidos foram ao encontro do que era esperado, tendo toda a solução sido validada pelo cliente. O mesmo destaca que a solução é rápida a devolver os resultados das análises efetuadas, como por exemplo Pivot Tables construídas sobre o modelo, cruzando várias dimensões e aplicando vários indicadores; os indicadores construídos apresentam os resultados corretos comparando com os mesmos implementados manualmente e o novo modelo implementado colmata as falhas identificadas nos modelos já existentes, como a desorganização dos elementos no modelo, a falta de robustez e a baixa escalabilidade.

Durante o desenvolvimento do projeto surgiram algumas dificuldades e limitações.

Uma das dificuldades identificadas deveu-se ao facto da fonte principal de dados residir em SAP e a mesma nunca ter sido utilizada anteriormente, o que requereu algum tempo adicional para um reconhecimento mais aprofundado da tecnologia. Outra dificuldade é relativa à arquitetura já montada para os modelos existentes, que apresentava algumas inconsistências e não era intuitiva, resultando também num esforço adicional de perceção da mesma. O mesmo aconteceu nos pacotes SSIS já existentes, que não eram claros quanto ao objetivo ou onde se encaixavam no processo ETL.

Quanto ao trabalho futuro, dado o estado atual da solução, o mesmo irá inicialmente focar-se em terminar a implementação dos dois restantes modelos analíticos - transportes e *Purchase to Pay* - de maneira a finalizar a solução analítica total, que desta forma disponibiliza para análise todos os dados que o cliente necessita. Adicionalmente, também é possível dar início a uma fase de construção de vários relatórios utilizando o Power BI, dinamizando e tornando mais apelativa a consulta feita aos dados dos modelos. A ferramenta apresenta ainda visualizações que utilizam inteligência artificial para calcular previsões de resultados, como por exemplo prever os gastos de um mês com base em meses anteriores e mostrar fatores de influência, o que será uma constante cada vez maior na área de análise de dados.

BIBLIOGRAFIA

- [1] M. Siegler. *Every 2 Days We Create As Much Information As We Did Up To 2003*. 2010. URL: <https://techcrunch.com/2010/08/04/schmidt-data/?guccounter=1>.
- [2] T. Olavsrud. *Executives still mistrust insights from data and analytics*. 2016. URL: <https://www.cio.com/article/3138049/executives-still-mistrust-insights-from-data-and-analytics.html>.
- [3] H. Moreno. *The Importance Of Data Quality – Good, Bad Or Ugly*. 2017. URL: <https://www.forbes.com/sites/forbesinsights/2017/06/05/the-importance-of-data-quality-good-bad-or-ugly/#c79c04a10c4d>.
- [4] D. Elias. *Conhecendo a arquitetura de Data Warehouse*. 2014. URL: <https://canaltech.com.br/business-intelligence/conhecendo-a-arquitetura-de-data-warehouse-19266/>.
- [5] T. Erl, W. Khattak e P. Buhler. *Big Data Fundamentals*. First. Prentice Hall, 2015. ISBN: 0134291077.
- [6] *Big Data What it is and why it matters*. URL: https://www.sas.com/en_us/insights/big-data/what-is-big-data.html.
- [7] G. Firican. *The 10 Vs of Big Data*. 2017. URL: <https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx>.
- [8] T. Shafer. *The 42 V's of Big Data and Data Science*. 2017. URL: <https://www.elderresearch.com/blog/42-v-of-big-data>.
- [9] C. Harvey. <https://www.datamation.com/big-data/big-data-pros-and-cons.html>. 2018. URL: [Big%20Data%20Pros%20and%20Cons](https://www.datamation.com/big-data/big-data-pros-and-cons.html).
- [10] *BI*. URL: <https://www.computer.org/csdl/magazine/co/2007/09/r9096/13rRUzpzefQ>.
- [11] *ETL*. URL: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/etl>.
- [12] *What is a Data Warehouse and Why Does It Matter To Your Business?* URL: <https://www.talend.com/resources/what-is-data-warehouse/>.
- [13] B. Stupakevich. *Top Five Benefits of a Data Warehouse*. 2011. URL: <https://www.smartdatacollective.com/top-five-benefits-data-warehouse/>.

- [14] S. George. *Inmon or Kimball: Which approach is suitable for your data warehouse?* 2012. URL: <https://www.computerweekly.com/tip/Inmon-or-Kimball-Which-approach-is-suitable-for-your-data-warehouse>.
- [15] *A tale of two architectures*. URL: <http://release.nl/magazines/Aveq/118597.pdf>.
- [16] W.H.Inmon. *Building the Data Warehouse*. First. Wiley, 2005. ISBN: 0764599445.
- [17] *Data Mart Tutorial: What is Data Mart, Types Example*. URL: <https://www.guru99.com/data-mart-tutorial.html>.
- [18] S. Rangarajan. *Data Warehouse Design – Inmon versus Kimball*. 2016. URL: <http://tdan.com/data-warehouse-design-inmon-versus-kimball/20300>.
- [19] *Facts Versus Dimensions Tables in a Database*. URL: <https://www.lifewire.com/facts-vs-dimensions-1019646>.
- [20] J. Wang e J. L. Kourik. “Data Warehouse Snowflake Design and Performance Considerations in Business Analytics”. Em: *Advances in Information Technology* 6.4 (2015). ISSN: 2315-4462.
- [21] *Online analytical processing (OLAP)*. URL: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/relational-data/online-analytical-processing>.
- [22] *Descrição geral dos cubos OLAP do Service Manager para análise avançada*. URL: <https://docs.microsoft.com/pt-pt/system-center/scsm/olap-cubes-overview?view=sc-sm-1807>.
- [23] *Cubo OLAP*. URL: <https://www.grapacity.com/en/blogs/working-with-olap-cube>.
- [24] R. Kimball e M. Ross. *The Data Warehouse Toolkit*. Third. Wiley, 2013. ISBN: 978-1-118-53080-1.
- [25] E. Drkušić. *Star Schema vs. Snowflake Schema*. 2016. URL: <https://www.vertabelo.com/blog/technical-articles/data-warehouse-modeling-star-schema-vs-snowflake-schema>.
- [26] *Download SQL Server Management Studio (SSMS)*. URL: <https://docs.microsoft.com/pt-pt/system-center/scsm/olap-cubes-overview?view=sc-sm-1807>.
- [27] *SQL Server Integration Services*. URL: <https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-2017>.
- [28] *About SQL Server Analysis Services*. URL: <https://docs.microsoft.com/en-us/sql/analysis-services/analysis-services?view=sql-server-2017>.
- [29] *Business intelligence like never before*. URL: <https://powerbi.microsoft.com/en-us/>.

- [30] *What is ERP and why do you need it?* URL: <https://dynamics.microsoft.com/en-us/erp/what-is-erp/>.
- [31] *SAP Order to Cash Process*. URL: <https://intrigosys.com/wp-content/themes/intrigo/images/sap-order-to-cash-process-graphic.png>.
- [32] *Star schema*. URL: <https://www.datawarehouse4u.info/Data-warehouse-schema-architecture-star-schema.html>.



CÓDIGO DE CONSTRUÇÃO DOS OBJECTOS ANALÍTICOS

```
1
2 ----- Criacao da vista da dimensao Material -----
3 SET ANSI_NULLS ON
4 GO
5
6 SET QUOTED_IDENTIFIER ON
7 GO
8
9
10 CREATE VIEW [ODS].[vwDimMaterial] AS
11
12
13 WITH
14 MATERIAL AS(
15 SELECT
16     m.MATNR      AS CodMaterial ,
17     m.MATKL      AS CodGrupoMercadoria ,
18     m.PRDHA      AS CodHierarquia ,
19     m.MTART      AS CodTipoMaterial ,
20     mdesp.MAKTX  AS DesMaterialPT ,
21     mdesE.MAKTX  AS DesMaterialEN ,
22     gmt.WGBEZ60  AS DesGrupoMercadorias ,
23     gmt.WGBEZ    AS DenominacaoGrupoMercadorias ,
24     tip.mtbez    AS DesTipoMaterial ,
25     m.MEINS      AS UnidadeMedidaBasica ,
26     dest.MATNR_CM AS MaterialDestino ,
27     max(zt.BUKRS) AS PrecoRevalorizado
28 FROM SAPABAP1.MARA m
```

ANEXO I. CÓDIGO DE CONSTRUÇÃO DOS OBJECTOS ANALÍTICOS

```

29 LEFT JOIN SAPABAP1.MAKT mdesp ON mdesp.MANDT = m.MANDT and mdesp.MATNR = m
.MATNR
30 LEFT JOIN (select distinct MATNR, MAKTX,MANDT from SAPABAP1.MAKT where
SPRAS = 'E') mdesE ON mdesE.MANDT = m.MANDT and mdesE.MATNR = m.MATNR
31 LEFT JOIN SAPABAP1.ZMM_REVAL_PRIC_T zt ON zt.MANDT=m.MANDT AND zt.MATNR=m.
MATNR
32 LEFT JOIN (SELECT MAX(MATNR_CM)MATNR_CM, MATNR FROM SAPABAP1.ZMM_MAT_CM_T
33 WHERE MANDT=100
34 GROUP BY MATNR) dest ON m.MATNR = dest.MATNR
35 LEFT JOIN SAPABAP1.T023T gmt ON gmt.MATKL = m.MATKL AND gmt.MANDT = m.
MANDT AND gmt.SPRAS = mdesp.SPRAS
36 LEFT JOIN SAPABAP1.T134T TIP ON TIP.MTART = M.MTART AND TIP.MANDT = m.
MANDT AND TIP.SPRAS = mdesp.SPRAS
37 WHERE m.MANDT=100 and mdesp.SPRAS IN ( 'P' )
38 group by
39 m.MATNR
40 ,m.MATKL
41 ,m.PRDHA
42 ,mdesp.MAKTX
43 ,mdesE.MAKTX
44 ,gmt.WGBEZ60
45 ,gmt.WGBEZ
46 ,m.MEINS
47 ,m.MTART
48 ,dest.MATNR_CM
49 ,tip.mtbez
50 )
51
52 ,HIERARQUIA_BASICA as (
53 select
54 hm.PRODH AS CodHierarquia ,
55 hmtpt.VTEXT AS DesHierarquiaMaterialPT ,
56 hmte.VTEXT AS DesHierarquiaMaterialEN ,
57 hm.STUFE AS Nivel
58 from SAPABAP1.T179 hm
59 left join SAPABAP1.T179T hmtpt on hm.PRODH = hmtpt.PRODH and hm.MANDT = hmtpt
.MANDT
60 left join (select distinct PRODH, VTEXT, MANDT from SAPABAP1.T179T where
SPRAS = 'E') hmte on hm.PRODH = hmte.PRODH and hm.MANDT = hmte.MANDT
61 where hm.MANDT='100 '
62 and hmtpt.SPRAS in ( 'P' )
63 )
64
65 , HIERARQUIA AS (
66 SELECT
67 CodHierarquia ,
68 DesHierarquiaMaterialPT ,
69 DesHierarquiaMaterialEN ,
70 Nivel ,
71 NULL AS CodHierarquiaCentroLucroPai

```

```

72 FROM HIERARQUIA_BASICA T1
73 WHERE Nivel = 1
74 UNION all
75 SELECT
76 T2.CodHierarquia ,
77 T2.DesHierarquiaMaterialPT ,
78 T2.DesHierarquiaMaterialEN ,
79 T2.Nivel ,
80 LEFT(T2.CodHierarquia ,LEN(X.CodHierarquia)) as
CodHierarquiaCentroLucroPai
81 FROM HIERARQUIA_BASICA T2
82 INNER JOIN HIERARQUIA_BASICA X ON T2.Nivel = 2 AND X.Nivel + 1 = T2.
Nivel AND X.CodHierarquia = LEFT(T2.CodHierarquia ,LEN(X.CodHierarquia))
83 UNION all
84 SELECT
85 T2.CodHierarquia ,
86 T2.DesHierarquiaMaterialPT ,
87 T2.DesHierarquiaMaterialEN ,
88 T2.Nivel ,
89 LEFT(T2.CodHierarquia ,LEN(X.CodHierarquia)) as
CodHierarquiaCentroLucroPai
90 FROM HIERARQUIA_BASICA T2
91 INNER JOIN HIERARQUIA_BASICA X ON T2.Nivel = 3 AND X.Nivel + 1 = T2.
Nivel AND X.CodHierarquia = LEFT(T2.CodHierarquia ,LEN(X.CodHierarquia))
92 UNION all
93 SELECT
94 T2.CodHierarquia ,
95 T2.DesHierarquiaMaterialPT ,
96 T2.DesHierarquiaMaterialEN ,
97 T2.Nivel ,
98 LEFT(T2.CodHierarquia ,LEN(X.CodHierarquia)) as
CodHierarquiaCentroLucroPai
99 FROM HIERARQUIA_BASICA T2
100 INNER JOIN HIERARQUIA_BASICA X ON T2.Nivel = 4 AND X.Nivel + 1 = T2.
Nivel AND X.CodHierarquia = LEFT(T2.CodHierarquia ,LEN(X.CodHierarquia))
101 UNION all
102 SELECT
103 T2.CodHierarquia ,
104 T2.DesHierarquiaMaterialPT ,
105 T2.DesHierarquiaMaterialEN ,
106 T2.Nivel ,
107 LEFT(T2.CodHierarquia ,LEN(X.CodHierarquia)) as
CodHierarquiaCentroLucroPai
108 FROM HIERARQUIA_BASICA T2
109 INNER JOIN HIERARQUIA_BASICA X ON T2.Nivel = 5 AND X.Nivel + 1 = T2.
Nivel AND X.CodHierarquia = LEFT(T2.CodHierarquia ,LEN(X.CodHierarquia))
110 )
111
112 SELECT cc.*

```

ANEXO I. CÓDIGO DE CONSTRUÇÃO DOS OBJECTOS ANALÍTICOS

```
113 ,COALESCE(h5.DesHierarquiaMaterialPT ,h4.DesHierarquiaMaterialPT ,h3.
DesHierarquiaMaterialPT ,h2.DesHierarquiaMaterialPT ,h1.
DesHierarquiaMaterialPT) AS Nivel1PT
114 ,CASE
115 WHEN h5.DesHierarquiaMaterialPT IS NULL AND h4.DesHierarquiaMaterialPT
IS NULL AND h3.DesHierarquiaMaterialPT IS NULL THEN h1.
DesHierarquiaMaterialPT
116 WHEN h5.DesHierarquiaMaterialPT IS NULL AND h4.DesHierarquiaMaterialPT
IS NULL THEN h2.DesHierarquiaMaterialPT
117 WHEN h5.DesHierarquiaMaterialPT IS NULL THEN h3.DesHierarquiaMaterialPT
118 ELSE h4.DesHierarquiaMaterialPT
119 END AS Nivel2PT
120 ,CASE
121 WHEN h4.DesHierarquiaMaterialPT IS NULL AND h3.DesHierarquiaMaterialPT
IS NULL THEN h1.DesHierarquiaMaterialPT
122 WHEN h5.DesHierarquiaMaterialPT IS NULL AND h4.DesHierarquiaMaterialPT
IS NULL THEN h1.DesHierarquiaMaterialPT
123 WHEN h5.DesHierarquiaMaterialPT IS NULL THEN h2.DesHierarquiaMaterialPT
124 ELSE h3.DesHierarquiaMaterialPT
125 END AS Nivel3PT
126 ,CASE
127 WHEN h5.DesHierarquiaMaterialPT IS NULL THEN h1.DesHierarquiaMaterialPT
128 ELSE h2.DesHierarquiaMaterialPT
129 END AS Nivel4PT
130 ,h1.DesHierarquiaMaterialPT AS Nivel5PT
131
132 ,COALESCE(h5.DesHierarquiaMaterialEN ,h4.DesHierarquiaMaterialEN ,h3.
DesHierarquiaMaterialEN ,h2.DesHierarquiaMaterialEN ,h1.
DesHierarquiaMaterialEN) AS Nivel1EN
133 ,CASE
134 WHEN h5.DesHierarquiaMaterialEN IS NULL AND h4.DesHierarquiaMaterialEN
IS NULL AND h3.DesHierarquiaMaterialEN IS NULL THEN h1.
DesHierarquiaMaterialEN
135 WHEN h5.DesHierarquiaMaterialEN IS NULL AND h4.DesHierarquiaMaterialEN
IS NULL THEN h2.DesHierarquiaMaterialEN
136 WHEN h5.DesHierarquiaMaterialEN IS NULL THEN h3.DesHierarquiaMaterialEN
137 ELSE h4.DesHierarquiaMaterialEN
138 END AS Nivel2EN
139 ,CASE
140 WHEN h4.DesHierarquiaMaterialEN IS NULL AND h3.DesHierarquiaMaterialEN
IS NULL THEN h1.DesHierarquiaMaterialEN
141 WHEN h5.DesHierarquiaMaterialEN IS NULL AND h4.DesHierarquiaMaterialEN
IS NULL THEN h1.DesHierarquiaMaterialEN
142 WHEN h5.DesHierarquiaMaterialEN IS NULL THEN h2.DesHierarquiaMaterialEN
143 ELSE h3.DesHierarquiaMaterialEN
144 END AS Nivel3EN
145 ,CASE
146 WHEN h5.DesHierarquiaMaterialEN IS NULL THEN h1.DesHierarquiaMaterialEN
147 ELSE h2.DesHierarquiaMaterialEN
148 END AS Nivel4EN
```



```

149 ,h1.DesHierarquiaMaterialEN AS Nivel5EN
150 FROM MATERIAL Cc
151 LEFT JOIN HIERARQUIA H1 ON Cc.CodHierarquia = h1.CodHierarquia
152 LEFT JOIN HIERARQUIA H2 ON h1.CodHierarquiaCentroLucroPai = h2.
CodHierarquia
153 LEFT JOIN HIERARQUIA H3 ON h2.CodHierarquiaCentroLucroPai = h3.
CodHierarquia
154 LEFT JOIN HIERARQUIA H4 ON h3.CodHierarquiaCentroLucroPai = h4.
CodHierarquia
155 LEFT JOIN HIERARQUIA H5 ON h4.CodHierarquiaCentroLucroPai = h5.
CodHierarquia
156
157
158
159 GO
160
161----- Criacao do procedimento -----
162
163 CREATE PROCEDURE [EDW].[uspDimMaterial]
164
165 AS
166 BEGIN
167
168 SET NOCOUNT ON
169
170
171 MERGE INTO [EDW].[DimMaterial] tgt
172 USING [AUX].[DimMaterial] src
173 ON
174 (
175     tgt.[CodMaterial] = src.[CodMaterial]
176 )
177 WHEN MATCHED
178 THEN
179 UPDATE
180 SET
181     tgt.[CodGrupoMercadoria] = src.[CodGrupoMercadoria]
182     ,tgt.[CodHierarquia] = src.[CodHierarquia]
183     ,tgt.[CodTipoMaterial] = src.[CodTipoMaterial]
184     ,tgt.[DesMaterialPT] = src.[DesMaterialPT]
185     ,tgt.[DesMaterialEN] = src.[DesMaterialEN]
186     ,tgt.[DesGrupoMercadorias] = src.[DesGrupoMercadorias]
187     ,tgt.[DenominacaoGrupoMercadorias] = src.[
DenominacaoGrupoMercadorias]
188     ,tgt.[DesTipoMaterial] = src.[DesTipoMaterial]
189     ,tgt.[UnidadeMedidaBasica] = src.[UnidadeMedidaBasica]
190     ,tgt.[MaterialDestino] = src.[MaterialDestino]
191     ,tgt.[PrecoRevalorizado] = src.[PrecoRevalorizado]
192     ,tgt.[Nivel1PT] = src.[Nivel1PT]
193     ,tgt.[Nivel2PT] = src.[Nivel2PT]

```

ANEXO I. CÓDIGO DE CONSTRUÇÃO DOS OBJECTOS ANALÍTICOS

```

194      ,tgt.[ Nivel3PT ]           = src.[ Nivel3PT ]
195      ,tgt.[ Nivel4PT ]           = src.[ Nivel4PT ]
196      ,tgt.[ Nivel5PT ]           = src.[ Nivel5PT ]
197      ,tgt.[ Nivel1EN ]           = src.[ Nivel1EN ]
198      ,tgt.[ Nivel2EN ]           = src.[ Nivel2EN ]
199      ,tgt.[ Nivel3EN ]           = src.[ Nivel3EN ]
200      ,tgt.[ Nivel4EN ]           = src.[ Nivel4EN ]
201      ,tgt.[ Nivel5EN ]           = src.[ Nivel5EN ]
202
203
204 WHEN NOT MATCHED THEN
205     INSERT
206     (
207         [ CodMaterial ]
208         ,[ CodGrupoMercadoria ]
209         ,[ CodHierarquia ]
210         ,[ CodTipoMaterial ]
211         ,[ DesMaterialPT ]
212         ,[ DesMaterialEN ]
213         ,[ DesGrupoMercadorias ]
214         ,[ DenominacaoGrupoMercadorias ]
215         ,[ DesTipoMaterial ]
216         ,[ UnidadeMedidaBasica ]
217         ,[ MaterialDestino ]
218         ,[ PrecoRevalorizado ]
219         ,[ Nivel1PT ]
220         ,[ Nivel2PT ]
221         ,[ Nivel3PT ]
222         ,[ Nivel4PT ]
223         ,[ Nivel5PT ]
224         ,[ Nivel1EN ]
225         ,[ Nivel2EN ]
226         ,[ Nivel3EN ]
227         ,[ Nivel4EN ]
228         ,[ Nivel5EN ]
229     )
230 VALUES
231     (
232         src.[ CodMaterial ]
233         ,src.[ CodGrupoMercadoria ]
234         ,src.[ CodHierarquia ]
235         ,src.[ CodTipoMaterial ]
236         ,src.[ DesMaterialPT ]
237         ,src.[ DesMaterialEN ]
238         ,src.[ DesGrupoMercadorias ]
239         ,src.[ DenominacaoGrupoMercadorias ]
240         ,src.[ DesTipoMaterial ]
241         ,src.[ UnidadeMedidaBasica ]
242         ,src.[ MaterialDestino ]
243         ,src.[ PrecoRevalorizado ]

```

```

244         ,src.[Nivel1PT]
245         ,src.[Nivel2PT]
246         ,src.[Nivel3PT]
247         ,src.[Nivel4PT]
248         ,src.[Nivel5PT]
249         ,src.[Nivel1EN]
250         ,src.[Nivel2EN]
251         ,src.[Nivel3EN]
252         ,src.[Nivel4EN]
253         ,src.[Nivel5EN]
254     );
255 END
256
257
258 ----- Criacao da factual de faturas
259 -----
260
261 CREATE VIEW [AUX].[vwFactFaturas] AS
262 SELECT
263     [DocFacturamento]
264     ,[CodItem]
265
266
267     --,[FactFaturas].[CodTipoDocFacturamento]
268     ,isnull(DimTipoDeFatura.[PKTipoDeFatura],-1) AS FKTipoDeFatura
269     --,[FactFaturas].[CodCategoriaDocumento]
270     ,isnull(DimCategoriaDocumento.[PKCategoriaDocumento],-1) AS
FKCategoriaDocumento
271     --,[FactFaturas].[CodOrganizacaoVendas]
272     ,isnull(DimOrganizacaoVendas.[PKOrganizacaoDeVendas],-1) AS
FKOrganizacaoDeVendas
273     --,[FactFaturas].[CodCanalDistribuicao]
274     ,isnull(DimCanalDistribuicao.PKCanalDistribuicao,-1) AS
FKCanalDistribuicao
275     --,[FactFaturas].[CodEmpresa]
276     ,isnull(DimEmpresa.PKEmpresa,-1) AS FKEmpresa
277     --,[FactFaturas].[CodEstadoDeLancamento]
278     ,isnull(DimEstadoDeLancamento.[PKEstadoDeLancamento],-1) AS
FKEstadoDeLancamento
279     --,[FactFaturas].[CodEstadoDaFatura]
280     ,isnull(DimEstado.[PKEstadoDocumento],-1) AS FKEstadoDaFatura
281     --,[FactFaturas].[CodMaterial]
282     ,isnull(DimMaterial.PKMaterial,-1) AS FKMaterial
283     --,[FactFaturas].[CodCentro]
284     ,isnull(DimCentro.PKCentro,-1) AS FKCentro
285     --,[FactFaturas].[CodCentroDeCusto]
286     ,isnull(DimCentroCusto.PKCentroCusto,-1) AS FKCentroCusto
287     --,[FactFaturas].[CodCentroDeLucro]
288     ,isnull(DimCentroLucro.PKCentroLucro,-1) AS FKCentroLucro

```

ANEXO I. CÓDIGO DE CONSTRUÇÃO DOS OBJECTOS ANALÍTICOS

```

289      --,FactFaturas.CodAreaContabilidadeCustos
290      ,isnull (DimAreaDeContabilidadeDeCustos.PKAreaDeContabilidadeDeCustos
    , -1) AS FKAreaDeContabilidadeDeCustos
291      --,FactFaturas.CodCondicaoExpedicao
292      ,isnull (DimExpedicao.PKExpedicao, -1) AS FKExpedicao
293      --,[CodSetorAtividade]
294      --,[CodTipoDocumento]
295      ,[CodDataCriacaoOrdemDeVenda]
296      ,[CodDataFaturamento]
297      ,[CodNumRemessa]
298      ,[CodItemRemessa]
299      ,[NumDocumento]
300      ,[Moeda]
301      ,[TaxaDeCambio]
302      ,[CondicaoPricing]
303      ,[QuantidadeFaturada]
304      ,[ValorLiquidoFatura]
305      ,[ValorLiquidoVisaoEmpLg]
306      ,[ValorLiquidoVisaoEmpCl]
307      ,[ValorLiquidoVisaoEmpGr]
308      ,[ValorLiquidoVisaoEurLg]
309      ,[ValorLiquidoVisaoEurCl]
310      ,[ValorLiquidoVisaoEurGr]
311      ,[ValorLiquidoVisaoTrxLg]
312      ,[MontImposto]
313      ,[UnidadeVenda]
314      ,[PesoLiquido]
315      ,[PesoBruto]
316      ,[UnidadeDePeso]
317      ,[Volume]
318      ,[UnidadeVolume]
319      ,[ValorLiquidoFatura1]
320      ,[ValorLiquidoFatura2]
321      ,[ValorLiquidoFatura3]
322      ,[ValorLiquidoFatura4]
323      ,[ValorLiquidoFatura5]
324      ,[ValorLiquidoFatura6]
325      ,[CustoTransporteReal]
326      ,[ValorLiquidoCIFVisaoEmpCl]
327      ,[ValorLiquidoCIFVisaoEmpGr]
328      ,[ValorLiquidoCIFVisaoEurLg]
329      ,[ValorLiquidoCIFVisaoEurCl]
330      ,[ValorLiquidoCIFVisaoEmpLg]
331      ,[ValorLiquidoCIFVisaoEurGr]
332      ,[ValorLiquidoCIFVisaoTrxLg]
333      ,[ValorLiquidoFOBVisaoEmpCl]
334      ,[ValorLiquidoFOBVisaoEmpGr]
335      ,[ValorLiquidoFOBVisaoEurLg]
336      ,[ValorLiquidoFOBVisaoEurCl]
337      ,[ValorLiquidoFOBVisaoEmpLg]

```

```

338      ,[ValorLiquidoFOBVisaoEurGr]
339      ,[ValorLiquidoFOBVisaoTrxLg]
340      ,[CustoVisaoEmpCl]
341      ,[CustoVisaoEmpGr]
342      ,[CustoVisaoEurLg]
343      ,[CustoVisaoEurCl]
344      ,[CustoVisaoEmpLg]
345      ,[CustoVisaoEurGr]
346      ,[CustoVisaoTrxLg]
347      ,[TipoDeInformacao]
348      ,DATEDIFF(day , [DataCriacaoOrdemDeVenda] , [DataFaturamento]) AS
TempoOVFatura
349
350 FROM [AUX].[FactFaturas] FactFaturas with(nolock)
351
352
353
354 LEFT JOIN [EDW].[DimCategoriaDocumento] DimCategoriaDocumento with(
nolock)
355 ON CAST(REPLACE(LTRIM(REPLACE( FactFaturas .[ CodCategoriaDocumento]
COLLATE SQL_Latin1_General_CP1_CS_AS , '0' , ' ')), ' ', '0') AS NVARCHAR(10))
= CAST(REPLACE(LTRIM(REPLACE( DimCategoriaDocumento .[ CodCategoriaDocumento]
COLLATE SQL_Latin1_General_CP1_CS_AS , '0' , ' ')), ' ', '0') AS NVARCHAR(10)
) -- Case Sensitive ON
356
357 LEFT JOIN [EDW].[DimMaterial] DimMaterial with(nolock)
358 ON CAST(REPLACE(LTRIM(REPLACE( FactFaturas .[ CodMaterial] , '0' , ' ')), ' ',
'0') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE( DimMaterial .CodMaterial
, '0' , ' ')), ' ', '0') AS NVARCHAR(10))
359
360 LEFT JOIN [EDW].[DimCentro] DimCentro with(nolock)
361 ON CAST(REPLACE(LTRIM(REPLACE( FactFaturas .[ CodCentro] , '0' , ' ')), ' ', '
0') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE( DimCentro .CodCentro , '0' ,
' ')), ' ', '0') AS NVARCHAR(10))
362
363 LEFT JOIN [EDW].[DimCentroCusto] DimCentroCusto with(nolock)
364 ON CAST(REPLACE(LTRIM(REPLACE( FactFaturas .[ CodCentroDeCusto] , '0' , ' '))
, ' ', '0') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE( DimCentroCusto .
CodCentroCusto , '0' , ' ')), ' ', '0') AS NVARCHAR(10))
365
366 LEFT JOIN [EDW].[DimCentroLucro] DimCentroLucro with(nolock)
367 ON CAST(REPLACE(LTRIM(REPLACE( FactFaturas .[ CodCentroDeLucro] , '0' , ' '))
, ' ', '0') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE( DimCentroLucro .
CodCentroLucro , '0' , ' ')), ' ', '0') AS NVARCHAR(10))
368
369 LEFT JOIN [EDW].[DimOrganizacaoVendas] DimOrganizacaoVendas with(nolock)
370 ON CAST(REPLACE(LTRIM(REPLACE( FactFaturas .[ CodOrganizacaoVendas] , '0' , '
')), ' ', '0') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE(
DimOrganizacaoVendas .[ CodOrganizacaoDeVendas] , '0' , ' ')), ' ', '0') AS
NVARCHAR(10))

```

ANEXO I. CÓDIGO DE CONSTRUÇÃO DOS OBJECTOS ANALÍTICOS

```

371
372     LEFT JOIN [EDW].[DimCanalDistribuicao] DimCanalDistribuicao with(nolock)
373     ON CAST(REPLACE(LTRIM(REPLACE(FactFaturas.[CodCanalDistribuicao], '0', '
    ')), ' ', '0') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE(
    DimCanalDistribuicao.CodCanalDistribuicao, '0', ' '), ' ', '0') AS NVARCHAR
    (10))
374
375     LEFT JOIN [EDW].[DimEmpresa] DimEmpresa with(nolock)
376     ON CAST(REPLACE(LTRIM(REPLACE(FactFaturas.CodEmpresa, '0', ' '), ' ', '0
    ') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE(DimEmpresa.CodEmpresa, '0
    ', ' '), ' ', '0') AS NVARCHAR(10))
377
378     LEFT JOIN [EDW].[DimAreaDeContabilidadeDeCustos]
    DimAreaDeContabilidadeDeCustos with(nolock)
379     ON CAST(REPLACE(LTRIM(REPLACE(FactFaturas.CodAreaContabilidadeCustos, '0
    ', ' '), ' ', '0') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE(
    DimAreaDeContabilidadeDeCustos.CodAreaContabilidadeDeCustos, '0', ' '), '
    ', '0') AS NVARCHAR(10))
380
381     LEFT JOIN [EDW].[DimTipoDeFatura] DimTipoDeFatura with(nolock)
382     ON CAST(REPLACE(LTRIM(REPLACE(FactFaturas.CodTipoDocFaturamento, '0', '
    ')), ' ', '0') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE(DimTipoDeFatura
    .[CodTipoDeFatura], '0', ' '), ' ', '0') AS NVARCHAR(10))
383
384     LEFT JOIN [EDW].[DimEstadoDocumentos] DimEstado with(nolock)
385     ON CAST(REPLACE(LTRIM(REPLACE(FactFaturas.[CodEstadoDaFatura], '0', ' '),
    ' ', '0') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE(DimEstado.[
    CodEstadoDocumento], '0', ' '), ' ', '0') AS NVARCHAR(10))
386
387     LEFT JOIN [EDW].[DimEstadoDeLancamento] DimEstadoDeLancamento with(nolock)
388     ON CAST(REPLACE(LTRIM(REPLACE(FactFaturas.[CodEstadoDeLancamento], '0', '
    ')), ' ', '0') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE(
    DimEstadoDeLancamento.CodEstadoDeLancamento, '0', ' '), ' ', '0') AS
    NVARCHAR(10))
389
390     LEFT JOIN [EDW].[DimExpedicao] DimExpedicao with(nolock)
391     ON CAST(REPLACE(LTRIM(REPLACE(FactFaturas.[CodCondicaoExpedicao], '0', ' ')
    )), ' ', '0') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE(DimExpedicao.
    CodCondicaoExpedicao, '0', ' '), ' ', '0') AS NVARCHAR(10))
392     AND CAST(REPLACE(LTRIM(REPLACE(FactFaturas.[CodLocalExpedicao], '0', ' '))
    , ' ', '0') AS NVARCHAR(10)) = CAST(REPLACE(LTRIM(REPLACE(DimExpedicao.
    CodLocalExpedicao, '0', ' '), ' ', '0') AS NVARCHAR(10))
393     AND '-2' = CAST(REPLACE(LTRIM(REPLACE(DimExpedicao.CodTipoExpedicao, '0',
    ' ')), ' ', '0') AS NVARCHAR(10))
394
395
396 GO
397
398

```